

Unicenter[®] CA-APAS[®] **Insight Monitor for Adabas**

User Guide

4.1



Computer Associates[®]

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation "as is" without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

© 2003 Computer Associates International, Inc.

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contents

Chapter 1: Introduction to Unicenter CA-APAS

What is Unicenter CA-APAS?	1-1
Comprehensive Facilities	1-1
Flexible Configuration	1-2
Unicenter CA-APAS Highlights	1-2
Command Logging Options	1-3
Derived Information	1-4
Efficiency of Unicenter CA-APAS	1-5
Performance History System	1-5
Tuning Guidelines	1-5
Using Unicenter CA-APAS to Tune Your Adabas Environment	1-6
Performance Analysis Tasks	1-6
Continuous Monitoring	1-7
Spot Checking	1-7
Crisis Resolution	1-7
Quality Assurance	1-8
Capacity Planning	1-8
Charge Back and Accounting	1-8
Audit Trail and Security	1-8
Debugging	1-9

Chapter 2: Using Unicenter CA-APAS Requests

Submitting Request Statements	2-2
Batch Mode Submission	2-2
MPM Mode Submission	2-2
Specifying the Data You Want	2-3
Data About Individual Commands	2-3
Data About Sets of Commands	2-4
Omission of BY Clause	2-4
Inclusion of BY Clause	2-5
Defining Your Own Data Fields	2-6
DECLARE Statement	2-6
Example:	2-6
USER-EXIT Routine	2-7
Limiting Data to Selected Commands	2-7
Selection Criteria	2-7
WHERE Clause Examples	2-8
Reporting Summary Data by Time Intervals	2-9
BY Clause	2-9
INTERVAL Parameter	2-10
Defining Outputs	2-11
Online Displays	2-11
Machine-Readable Data Files	2-11
Print-Formatted Reports	2-12
Field Length and Format	2-12
Column Headings	2-12
Stacking Fields in a Column	2-13
Command Log Records	2-14
Dual Files	2-14
Request Security	2-14
OWNER Values	2-14
SECURITY Values	2-15

Chapter 3: Specific Techniques for Selected Objectives

Getting Started	3-1
Exception Reporting of Selected Commands	3-2
Commands Which Exceed Performance Limits	3-2
Commands Which Receive Exceptional Response Codes	3-3
Commands Referencing Sensitive Files	3-4
Commands Whose Use is to be Restricted	3-5
Analyzing Usage of Adabas by a Program	3-5
Program Trace	3-5
Program Profile	3-6
Analyzing Search Logic	3-7
Investigating Hold Queue Overflow	3-8
Analyzing Hold Logic	3-9
Analyzing Peak Processing Times	3-9
Sizes of Time Periods	3-10
Choosing Between BY and INTERVAL	3-10
The Fields You Report	3-10
Determining Major Sources of Workloads	3-11
Choice of Summary Function	3-11
Detecting Adabas Processing Bottlenecks	3-12
Update Commands	3-12
Complex Searches	3-12
Shortage of Threads	3-12
Analyzing Impact of Batch Jobs on Response Times	3-13
Determining CPU-Bound versus I/O-Bound	3-14
Identifying Candidates for Prefetch/Multifetch	3-14
Evaluating Effectiveness of ADAM	3-15
Optimizing Logical Sequential Processing	3-15

Chapter 4: Using Unicenter CA-APAS Online

Overview	4-1
Unicenter CA-APAS Capabilities	4-2
Conducting a Unicenter CA-APAS Insight Session.....	4-3
Basic Procedures.....	4-3
Initial Screen	4-3
Entering Unicenter CA-APAS Insight Commands and Operand	4-4
Entering Natural Commands and Operands	4-5
Using Unicenter CA-APAS Insight HELP	4-5
Continuing an Interrupted Unicenter CA-APAS Insight Session	4-5
Terminating a Unicenter CA-APAS Insight Session.....	4-6
Manipulating Unicenter CA-APAS Requests	4-7
Automatic Generation and Activation of New Requests	4-7
Selection of Saved Requests	4-8
Starting Saved Requests	4-8
Preparing Unicenter CA-APAS Requests for Processing	4-9
Syntax Checking and Activating Requests	4-9
Resetting Request Data	4-9
Suspending and Restarting Requests.....	4-10
Purging Requests from the Data Collector	4-10
Showing Request Status	4-11
Predefined Requests	4-11
Displaying Request Data	4-12
Standard Functioning of the DISPLAY Command	4-12
Special DISPLAY Option	4-12
Scrolling Data	4-13
Scrolling Procedure	4-13
Logic and Considerations.....	4-14
SNAP Display of Adabas Nucleus Activity	4-14
Switching Dual Output Files.....	4-15
Turning CA-SpaceMan I/O Logging On or Off	4-15
Unicenter CA-APAS Zap Status.....	4-16
Adabas Functions	4-16
Resolving Errors	4-16
Errors Reported by Natural	4-16
Syntax Errors in Unicenter CA-APAS Requests	4-17
Invalid Commands or Operands	4-17
Scrolling Errors	4-17
Terminal Screen Size Errors.....	4-18
Data Collector Abnormal Terminations	4-18
Other Errors.....	4-18

Effective Use Of Unicenter CA-APAS Requests	4-19
Report Dimensions	4-19
Report Line Width	4-19
Quantity of Report Lines	4-20
Request Objectives	4-21
Determining Causes of Current Poor Performance	4-21
Exception Monitoring	4-22
Useful Request Options and Fields	4-22

Chapter 5: Batch Execution of Unicenter CA-APAS

Supported Commands	5-1
--------------------------	-----

Chapter 6: Performance History System

Areas of Application	6-1
Components	6-2
Inputs	6-2
Command Data	6-2
User-Supplied Data	6-3
History File	6-3
History Programs	6-3
Sample Inquiries	6-4
Hourly Workload Panel	6-4
Five Heaviest Adabas File Numbers Panel	6-5
Five Heaviest Adabas Batch Jobs Panel	6-5
Five Heaviest Natural User IDs Panel	6-5
Five Heaviest Natural User Programs Panel	6-6

Appendix A: Sample Unicenter CA-APAS Insight Session

Invoking Unicenter CA-APAS Insight from Natural	A-1
Initial Insight Panel	A-2
Main Menu Panel	A-3
SELECT Panel	A-4
SELECT Panel - After Browse	A-5
SHOW Panels	A-6
FILLOAD Panel	A-7
Generate Trace Request Panel - 1 of 2	A-8
Generate Trace Request Panel - 2 of 2	A-9
Natural Editor Panel	A-10
Activate the NATLOAD Request	A-12
NATLOAD Panel	A-12
Deleting Requests	A-13
Creating A New Request	A-15
Syntax Checking the New Request	A-17
Correcting a Syntax Error	A-18
Displaying a New Request	A-19
SNAPSHOT Panel	A-20
Insight Help Main Menu	A-21
SNAP Display Help Panel Menu	A-22
SNAP Display Help Detail Panel	A-23
ADAMENU - The Adabas Functions Menu	A-25
High Water Marks Panel	A-26
User Queue Panel	A-27
Hold Queue Summary Screen	A-28
I/O Buffer Pool Panel	A-29
Stop Users Panel	A-30
Operator Command Panel	A-31

Glossary

Index

Introduction to Unicenter CA-APAS

This chapter provides an overview of Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS) concepts and facilities. The concepts described in this chapter are fundamental to your understanding of the power and flexibility of Unicenter CA-APAS.

What is Unicenter CA-APAS?

Unicenter CA-APAS is the Computer Associates Adabas Performance Analysis System, a comprehensive set of software components that enable you to more effectively manage Adabas components and the applications serviced by the Adabas nucleus. Unicenter CA-APAS is a software tool to monitor and tune your Adabas environment and alert you to potential and ongoing performance problems and bottlenecks.

Unicenter CA-APAS collects data from Adabas Command Log records; from information Unicenter CA-APAS collects from the command application region/partition, and directly from the Adabas control blocks, nucleus and threads. The staff in your data center is responsible for the performance and management of Adabas and can use this data to effectively manage your Adabas environment.

Comprehensive Facilities

Unicenter CA-APAS includes a wide range of facilities for managing performance in Adabas environments:

- Extensive background information about Adabas performance is provided to increase the expertise of personnel responsible for Adabas operations and applications.
- Important performance data items not otherwise available are captured and correlated with standard Adabas data items.
- Default and user-defined exception and summary reports make essential information continually available to database administration personnel.

- A flexible and powerful real-time capability enables rapid response to unanticipated problems as they are happening; it also allows online quality assurance measurement and debugging of applications being developed.
- The Performance History System enables development of a base of information that supports retrospective analysis and projections of future loads and capacity.
- Unicenter CA-APAS is more than a mere collection of useful tools; it provides capabilities that are essential to effective database performance management.

With all components of Unicenter CA-APAS installed, the information you need is readily available to support tasks ranging from resolution of the most urgent real-time crisis to long-range planning for a year or more into the future. Database administrators and others responsible for database performance can then work with the lights turned on.

Flexible Configuration

Unicenter CA-APAS provides the power and flexibility to grow with the needs of your organization. As soon as you load the distribution tape, you can begin generating valuable information with no changes in current operational procedures. We provide sample JCL to execute the Data Collector as a batch job to read Adabas Command Log files and produce useful reports defined by default requests.

You can begin to use the more powerful and sophisticated features of Unicenter CA-APAS in increments that fit your requirements, resources, and interests. For more information about progressive use of Unicenter CA-APAS features, see the chapter “Functional Components” in the *Unicenter CA-APAS Systems Guide*.

Unicenter CA-APAS Highlights

The Insight component of Unicenter CA-APAS allows you to control the full range of data gathering and output capabilities of Unicenter CA-APAS in a dynamic, interactive mode from online terminals.

All types of Unicenter CA-APAS requests for detail or summary reports can be created, modified, syntax checked, started, paused, resumed, and deleted from terminals.

Current results from detail and summary reports can be displayed at terminals, and a special set of data about Adabas nucleus activity and status can be displayed.

Command Logging Options

Many organizations are reluctant to write Adabas Command Logs because of the performance overhead and the difficulty of managing the large number of Command Log files involved. In addition, it is often difficult to interpret or relate data to a particular application.

Unicenter CA-APAS offers improvements over the standard Adabas command logging process.

1. Unicenter CA-APAS permits the physical writing of Adabas Command Log files to be totally *eliminated* without any loss of reporting capability. All of the real-time terminal displays, printed reports and data files possible through Unicenter CA-APAS can be produced by the Data Collector running with Adabas. In this mode, Adabas passes Command Log record images to the Data Collector but does not physically write them to an external file. This reduces Adabas I/O overhead significantly. It also eliminates the procedural problems of dual or multiple command logging and management of generations of Command Log files on tape.
2. Unicenter CA-APAS offers a more efficient and flexible alternative to Adabas command logging. COPY requests cause the Data Collector to write Command Log records supplemented with the additional data Unicenter CA-APAS derives to Unicenter CA-APAS COPY files. Unicenter CA-APAS I/O does not delay Adabas processing. COPY requests can be controlled online through Insight to dynamically change command selection criteria and choice of buffers being logged.
3. Unicenter CA-APAS offers a simple and flexible way to do *selective* Adabas command logging. LOG and UNLOG statements can be used to specify command selection criteria to control which commands, if any, Adabas writes to its Command Log files. LOG and UNLOG statements can be submitted online through Insight.

In summary, Unicenter CA-APAS offers full capability for selecting and reporting command data from any combination of:

- Command Log record images passed to the Data Collector during an active Adabas session
- Stand-alone batch processing of Command Log files written earlier by Adabas
- Stand-alone batch processing of COPY files written earlier by the Unicenter CA-APAS Data Collector

Derived Information

A number of data fields very useful for Adabas performance analysis and tuning are not available in the standard Command Log records. Unicenter CA-APAS makes these fields available for use in any combination with the standard fields. Some important derived fields are:

- Natural Library and Program Name: Essential for tuning applications written in Natural.
- Command Enqueue Time: The delay between the time an application issues a call and the time Adabas starts processing the command in one of its threads. Essential for detecting bottlenecks in processing update or complex search commands and for isolating Adabas processing time from TP monitor processing time when investigating reasons for slow response.
- TP Transaction Count for CICS: Allows estimates of number of Adabas calls per individual execution of a CICS transaction by users.
- Descriptor Field Usage: Reports on the frequency with which descriptors are actually used in search criteria and for logical sequential processing.
- Number of Records Held by a Program: Helps spot faulty hold logic that could cause Hold Queue overflows or transaction time-outs.
- Breakdown of I/O by User-defined RABN Ranges: Assists in tasks such as balancing loads across physical units, analyzing activity against the three parts of the WORK data set, or analyzing activity against individual descriptor index blocks.
- User-defined Fields: Fields derived according to user-specified rules.
- Estimated CPU: Based on Software AG formulas with provision for user adjustment.
- Command Cost: Provides information for charge back based on CPU and I/O.
- Buffer Flush Counts: Important in assessing buffer pool efficiency.

Efficiency of Unicenter CA-APAS

Compared to other methods, Unicenter CA-APAS provides major economies in processing command data. Special algorithms greatly reduce CPU time from that used by a general purpose report-writer software or even other specially written programs.

Specifically, Data Collector components use the following techniques to optimize efficiency:

- All modules are coded in assembler.
- Summary tables are built using a special hashing algorithm.
- Selection criteria and user-defined field definitions are compiled rather than interpreted.
- No data accumulation or output logic is executed for commands that fail the selection criteria of the request.

The Data Collector can be executed as a sub-task rather than as a simple subroutine, thus it does its processing only when Adabas (the main task) has no need for CPU time. The result is little or no impact on Adabas throughput.

Performance History System

Unicenter CA-APAS includes a Performance History System that provides easy storage and access to commands summary and nucleus/utility session statistics in a single Adabas file. The Performance History System enables rapid review of Adabas loads and performance characteristics across combinations of parameters and entities such as:

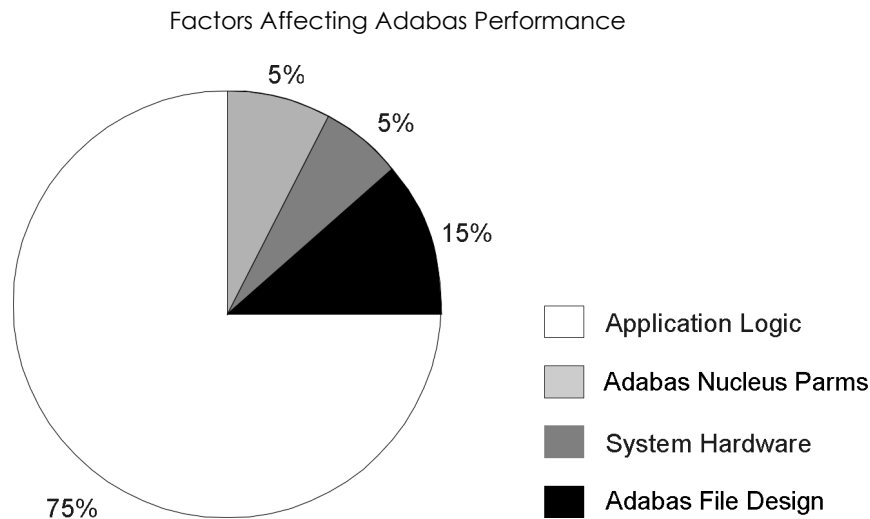
- User-specified time period (hour, day, week or month).
- CPU-ID and DBID or SMPID combination.
- Adabas file, command code.
- JOBNAME, TERMINAL-ID, NATURAL-ID.
- CICS transaction code.

Tuning Guidelines

The *Unicenter CA-APAS User Guide* is a valuable training aid for users new to Adabas. This guide shows how to use information collected by Unicenter CA-APAS to tune the Adabas nucleus parameters, database design and configuration, and application programs. We identify common performance problems and suggest ways to solve these problems.

Using Unicenter CA-APAS to Tune Your Adabas Environment

Unicenter CA-APAS produces information that can be used to perform critical tasks in many areas. Most organizations use Unicenter CA-APAS primarily for performance analysis and tuning. Managing Adabas performance means planning for long-range growth of Adabas use, assuring that new systems meet performance objectives and reacting to short-term problems in a timely fashion. The factors that influence Adabas performance, along with their relative impacts, can be viewed as follows:



Performance Analysis Tasks

Use Unicenter CA-APAS for performance analysis tasks such as:

- Identifying application programs that are adversely affecting the performance of the entire system, or that are not accessing the database optimally.
- Properly setting Adabas nucleus parameters and determining the effect on performance of changing these parameters.
- Identifying periods of poor response and discovering responsible factors within the Adabas environment, concluding that the poor responses are due to factors other than Adabas processing.
- Determining the load placed on the system by various applications and balancing loads by scheduling the applications optimally.

Continuous Monitoring

Unicenter CA-APAS can routinely generate a number of exception reports, summary reports, and output files.

The exception reports, scanned daily, pinpoint applications that are generating overly expensive commands. This is especially important where ad hoc usage is common, or where the DBA has little control over application code quality. In addition, selected response codes (particularly those which indicate shortage of Adabas work area space) should be reviewed.

Spot Checking

During early stages of Unicenter CA-APAS use, and after any major change in the usage configuration of Adabas, Unicenter CA-APAS may be used to monitor Adabas nucleus statistics and overall Adabas usage. This real-time look gives an invaluable feel for how Adabas works in different situations.

In addition, non-routine printed reports may be generated periodically to determine whether changes have occurred in selected aspects of Adabas processing.

Crisis Resolution

In even the most well planned, well-managed environment, users sometimes complain that response is exceptionally bad or operations personnel report unusual occurrences.

Unicenter CA-APAS is useful for immediate diagnosis of such problems. By examining nucleus statistics and the standard summary and exception information constantly being generated by Unicenter CA-APAS, the type of problem can be quickly identified.

After the type of problem is identified, special displays for a particular job, terminal, system or program may be activated to identify the specific causes of the problem.

The technical solution to the problem will emerge rapidly from this dynamic, real-time investigation.

Quality Assurance

New systems and programs can be checked for efficient use of Adabas by obtaining detailed traces for a single terminal testing the system. Inefficient Adabas calls and excessive numbers of calls can be identified prior to performance surprises when a system is put into production.

Capacity Planning

In standard capacity planning, two components are necessary:

- Tracking growth of existing applications.
- Assessing the impact of new applications.

The Unicenter CA-APAS Performance History System gives immediate output showing the growth trends of existing systems. Trends can be tracked on both an overall basis and, more importantly, at peak load periods.

In assessing the impact of new applications, the Performance History System allows the DBA to find applications with load characteristics similar to the new one and to look at the resource requirements of these applications.

Finally, the Adabas nucleus statistics and overall usage data allow the DBA to assess the excess capacity of an Adabas nucleus and decide when Adabas work should be spread across several copies of the nucleus.

Charge Back and Accounting

Use Unicenter CA-APAS for charge back and accounting tasks such as:

- Calculating charges for individual commands based on I/O counts and estimated CPU usage.
- Aggregating command charges to defined user entities.

Audit Trail and Security

Use Unicenter CA-APAS for audit trail and security tasks such as:

- Logging all Adabas security violation response codes.
- Recording all accesses to sensitive files.
- Logging execution of all or selected batch jobs and terminal access to the database or selected files.

Debugging

Use Unicenter CA-APAS for debugging tasks such as:

- Logging all unusual Adabas response codes, including those indicating internal errors in the database, inadequate sizes of Adabas nucleus tables, or application errors in Adabas calls.
- Tracing commands issued by application programs to identify program logic errors.
- Logging Adabas buffer data for specific programs for analysis of search criteria or format specifications.

Using Unicenter CA-APAS Requests

Almost all Adabas performance data from Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS) is produced in response to specifications given in Unicenter CA-APAS request statements. This section suggests ways to write request statements to get the data you want about commands the Adabas nucleus processed during a given session.

Learning about command data requests is best done by referring to the following documents in the order shown:

1. For an introductory overview of command data requests, see the chapter “Introduction to CA-APAS Requests” in the *Writing Requests* guide.
2. This section of the *User Guide* explains the basics of writing command data requests to specify the kinds of reports, displays, or machine-readable outputs you want. For specific approaches to getting information about quite a few specific performance issues, see the following chapter “Specific Techniques For Selected Objectives.”
3. The Unicenter CA-APAS Insight library in a Natural system file and the Unicenter CA-APAS SOURCE library contain sample request statements. It is useful to refer to those sample requests in connection with reading these chapters of the *User Guide*. You might also find it convenient to edit copies of some of the sample requests to develop requests that satisfy your particular needs.
4. *Writing Requests* provides detailed statement syntax rules and field descriptions that you will need to consult occasionally as you write request statements.

Note: If you have not already read the “Introduction to Unicenter CA-APAS Requests” chapter in *Writing Requests*, please do so now before proceeding in this section.

Most of the Adabas performance data you can obtain from Unicenter CA-APAS is developed from or in connection with Adabas Command Log records. Request statements tell the Data Collector which Command Log records to process, what data items you want, and how the data should be output.

The basic steps in preparing and processing requests are described in this chapter.

Submitting Request Statements

The Data Collector can be executed in either of two modes:

- Batch mode executes as a stand-alone batch job and reads Command Log records from a physical file that was written earlier in a separate execution of Adabas
- MPM mode executes within the MPM job and processes Adabas Command Log record images as they are passed to it in main memory from Adabas

For more information about the two modes of execution, see the chapter “Functional Components” in the *Systems Guide*.

Batch Mode Submission

You submit request statements to the Data Collector as part of its control statement input stream when it is executed in batch mode. The statements must be created and placed in the appropriate file prior to the job being started.

If you submit the job, you can set up the input stream yourself. If the database administrator or someone else submits the job, you'll need to have that person add your statements to the input stream.

Specifications for the contents of the input stream are given in the “General Syntax Information” chapter of *Writing Requests* guide. For a description of how the input stream is specified in your operating system, see the chapter “Unicenter CA-APAS Data Collector” in the *Systems Guide*.

MPM Mode Submission

When the Data Collector is executed in MPM mode, your request statements can be submitted in either or both of two ways.

- They can be included in the input stream as for batch mode. This should be coordinated with the person who is responsible for executing the MPM.
- You can submit requests from an online terminal through Insight at anytime during the MPM session. There are several ways to do this:
 - Use the Insight SELECT function to list names of saved requests and mark one to be started
 - Use the Insight START command to specify a saved request to start
 - Use the Natural READ or EDIT command to place a request into the source work area, then execute the Insight ADD command

For more detailed information about these methods, see the chapter “Using Unicenter CA-APAS Online.” They are also shown in the appendix “Sample Unicenter CA-APAS Session” of this guide.

You can also DELETE, RESET or PAUSE/RESUME requests through Insight. These operations are applicable for both requests started from the input stream and those started online with Insight.

Specifying the Data You Want

In each request you can choose which command data fields or summary functions you want reported. You can also define the set of commands from which the data will come. The basics of doing this are described below. For detailed information and lists all of the data fields and summary functions that are available, see the *Writing Requests* guide.

Data About Individual Commands

Use an EXTRACT statement when you want to get detailed information about individual commands. Any of the command data fields listed in *Writing Requests* guide as well as any fields defined earlier in DECLARE statements may be included in the field list part of an EXTRACT statement. None of the summary functions are permitted since each data value from an EXTRACT pertains to one particular command.

You can use the EXTRACT statement to:

- Do exception reporting of individual commands
- Trace the chronological sequence of any set of commands
- See what control block and buffer values are being passed from application programs to Adabas and vice versa
- See the performance characteristics of an individual command

Exception reporting is discussed later in this section. A simple EXTRACT request and its output are shown next.

```
ALLCMDS: EXTRACT
FILE COMMAND-CODE RESPONSE-CODE ENQ-TIME DURATION
CPU JOBNAME TERMINAL-ID NAT-MOD-ID
INSIGHT-LINES = 20;
```

FIL NUM	C M D	RSP	ENQTIME SECS	DURAT SECS	CPU SECS	JOB NAME	TERM ID	NATURAL MODULE ID
51	L1	0	0.00002	0.00001	0.001	CICDEVDO	S282	SMIS WA3A509HM00010
51	L1	0	0.00002	0.00000	0.001	CICDEVDO	S282	SMIS WA3A509HM00010
51	L1	0	0.00002	0.00001	0.001	CICDEVDO	S282	SMIS WA3A509HM00010
48	S1	0	0.00002	0.00002	0.000	CICDEVDO	S282	SMIS WA3A509HM00010
48	RC	0	0.00002	0.00000	0.000	CICDEVDO	S282	SMIS WA3A509HM00010
51	S1	0	0.00002	0.00001	0.000	CICDEVDO	S282	SMIS WA3A509HM00010
51	RC	0	0.00002	0.00000	0.000	CICDEVDO	S282	SMIS WA3A509HM00010
51	L1	0	0.00002	0.00001	0.001	CICDEVDO	S282	SMIS WA3A509HM00010

Data About Sets of Commands

Use a SUMMARIZE statement to obtain information that applies to a set of commands as a whole rather than being unique to individual commands.

Following the SUMMARIZE keyword and any of its optional parameters, write the names of the summary functions you want for the set of commands. Write only summary functions, as shown in *Writing Requests* guide, at this point in the statement. Do not use any of the command data fields here.

Omission of BY Clause

Some summary functions, such as COUNT and JOB-ELAPSED-TIME, are completely specified just by specifying their names. Others, such as SUM and MIN, require the addition of a data field name in parentheses. Omit a BY clause if you only want the single set of summary function values that reflect the entire set of commands being reported.

Following is an example of this type of request and its output.

```
TOTLOAD: SUMMARIZE
COUNT SUM(TOTAL-IO) MAX(TOTAL-IO) MEAN(TOTAL-IO)
INSIGHT-LINES = 20;
```

CMD COUNT	SUM TOT IO	MAX TOT IO	MEAN TOT IO
752	230	36	0.91

Inclusion of BY Clause

If you want sets of summary function values reported for subsets of commands, then use a BY clause. Only command data fields can be used in a BY clause, not summary functions. Values for the BY fields are reported, but only to identify each of the subsets of commands for which summary function values are developed.

For example, adding a BY clause to the previous example results in the following request and output.

```
FILLOAD: SUMMARIZE
COUNT SUM(TOTAL-IO)
MAX(TOTAL-IO) MEAN(TOTAL-IO)
BY FILE
INSIGHT-LINES = 20;
```

FIL NUM	CMD COUNT	SUM	MAX	MEAN
		TOT IO	TOT IO	TOT IO
0	6	33	33	5.50
1	13	3	3	0.23
2	18	3	3	0.17
3	5	3	3	0.60
7	3	5	3	1.67
10	72	40	4	0.56
46	8	11	5	1.38
245	14	5	2	0.36
-----		-----		
	139	103	33	0.74

Be careful when deciding which and how many fields to include in a BY clause. Each unique combination of values that occurs for the set of BY fields requires its own storage area in which to accumulate the summary values that have been requested. The amount of virtual storage the Data Collector requires and the quantity of data generated for reading or post-processing increases in proportion to the number of unique BY levels. Please review the discussion of BY levels in *Systems Guide*.

If you include a BY clause in a SUMMARIZE statement, then decide whether you want summary function values developed only at the lowest level of command subsets, or whether you want subtotals and grand-totals developed for aggregations of the subsets of commands. To suppress these additional levels of data, include the parameter, SUBTOT=OFF, following the SUMMARIZE keyword.

Defining Your Own Data Fields

You can define new data fields in the two ways described below.

DECLARE Statement

One way is to write and submit a DECLARE statement to define a new field.

In the DECLARE statement you either specify the redefinition of all or part of another field, or you provide logic for assigning values to the new field based on tests of the values of other fields.

Example:

```
DECLARE JOBCODE (A4 HD='JOB','CODE')
  REDEFINES JOBNAME(4,3); /* CHARS 3-6
  DECLARE DEPT (N3 HD='DEPT','NO') VALOF
  IF JOBCODE <= 'C104' RESULTIS 201
  ELSEIF JOBCODE > 'C104' AND <= 'H210' RESULTIS 460
  ELSEIF JOBCODE > 'H210' AND <= 'Q360' RESULTIS 600
  ELSEIF JOBCODE = 'CICS' THEN
    IF TERMID(1) = 'K' THEN RESULTIS 801
    ELSEIF TERMID(1) = 'L' THEN RESULTIS 802
    ELSE RESULTIS 899
  IFEND
  ELSE RESULTIS 999
IFEND
```

The DECLARE statement that defines a new field must itself have been successfully syntax checked and activated by the Data Collector before the name of the new field is accepted in other statements.

Once accepted, declared field definitions remain in effect for the remainder of the current execution of the Data Collector. The declared field name can be used by any request. DECLARE statements cannot be deleted.

DECLARE specifications are not saved by one execution of the Data Collector for use in future executions. DECLARE statements must be submitted anew in each Data Collector execution in which the declared fields are needed.

USER-EXIT Routine

The other way is to provide a routine that is called from a Data Collector user-exit.

The routine can test other fields and then assign values to one or more of six special fields that are reserved for this purpose. These fields can be referenced in command data requests. The six fields are described in the *Writing Requests* guide.

Their names are:

- USERA, USERB, USERC for alphanumeric values
- USER1, USER2, USER3 for numeric values

Specifications for writing and invoking user-exit routines are given in *Systems Guide*. A sample routine is included in the Unicenter CA-APAS SOURCE library. Coordinate with your database administrator regarding user exits.

Limiting Data to Selected Commands

Use a WHERE clause in any EXTRACT, SUMMARIZE or COPY statement to control which Adabas commands are processed by the statement. COPY requests are described below.

If a statement contains a WHERE clause, only those commands which satisfy the logical conditions of the WHERE are processed by the statement. Without a WHERE clause, all commands received by the Data Collector while the statement is active are processed by the statement. For detailed information about the rules for writing the logical expressions of WHERE clauses and for definitions of which fields can be referenced, see the *Writing Requests* guide.

Selection Criteria

Selection criteria typically refer to fields such as:

- Identifications of entities like Adabas file number (FILE), Natural library/application (NAT-LOGON), Natural program (NAT-PROGRAM), main program (PROGRAM), job (JOBNAME), terminal (TERMINAL-ID) or user (USER-ID)
- Time of day
- Processing times such as length of time the command spent in the command queue waiting to be started in an Adabas thread (ENQ-TIME), duration in the thread (DURATION), or estimated CPU time (EST-CPU-TIME)

- Number of Adabas physical I/Os to Work (WORK-IO), Data (DATA-IO), Associator (ASSO-IO) or total I/O (TOTAL-IO)
- Particular values of fields such as COMMAND-CODE, COMMAND-ID, COMMAND-OPT1, COMMAND-OPT2, or password (ADDITIONS-3)

WHERE Clause Examples

In the following example, only commands to files 10, 20, or 30 are processed by the statement.

```
... WHERE FILE = 10 OR = 20 OR = 30...
```

In the next example, commands issued from any job named CICSTEST are excluded from processing by the statement.

```
... WHERE JOBNAME NOT EQUAL 'CICSTEST'
```

Be careful to express your intended meanings correctly when specifying multiple or complex selection criteria. This is important anytime you write AND, OR, NOT or parentheses in your logical criteria. To illustrate,

```
... WHERE FILE NOT EQUAL 100 OR FILE NOT EQUAL 200...
```

includes all commands, regardless of the file number (file 100 fails the first condition but satisfies the second, etc.) and thus gives the same result as if no WHERE had been included, while

```
... WHERE FILE NOT EQUAL 100 AND FILE NOT EQUAL 200...
```

excludes any commands to file 100 and excludes any commands to file 200.

Reporting Summary Data by Time Intervals

There are two ways you can have summary data developed for time intervals. The two methods differ in output format and how the Data Collector manages the summary tables involved.

BY Clause

The first way is to specify a field such as QUARTER-HOUR or HOUR as a BY field.

This approach limits you to those particular interval sizes. The intervals are reported by number, that is, hours 1 through 24 or quarter-hours 1 through 96, with the numbering always starting as of the immediately preceding midnight. Because the intervals are reported by number rather than time of day, it helps to write MIN(TIME) and/or MAX(TIME) as the very first item(s) in the summary function list of the statement. This makes it easier to associate each numbered interval with its time of day.

Advantages of this method over using an INTERVAL parameter, which is described next, are:

- You can see data for many time periods on a single report page or terminal display
- The SORT DESC option can report heaviest-use time periods at the beginning of the report or output file instead of listing them in chronological order

Potential disadvantages are:

- Increased use of virtual storage because the data for the entire report must be retained in virtual storage until the MPM session terminates or until you delete the request
- Any time periods in which there were no Adabas commands finished would not be reported
- With the exception of output to online display, abnormal termination would result in loss of the entire summary table before it had been output

INTERVAL Parameter

The other way is to include an INTERVAL parameter in the statement.

This approach lets you define the size of the interval in time units such as seconds, minutes or hours. The Data Collector decides when the first interval to be reported started. To do this, it begins at the immediately preceding midnight and comes forward in increments of the specified interval.

At the end of each interval, the values in the summary table are output, the table is reset to zeroes, and summarization then begins for the next interval.

This kind of processing has the following potential advantages over the BY processing described above.

- All intervals are reported, even those during which there were no commands
- Less data is at risk of being lost in the event of abnormal termination
- Less virtual storage is needed for the summary table

Defining Outputs

Several types of output can be obtained from command data requests. An `EXTRACT` or `SUMMARIZE` statement can produce any combination of the following kinds of outputs, each containing the specified data fields or summary functions:

- Online displays for viewing through Unicenter CA-APAS Insight
- Reports formatted for printing
- Machine-readable data files

A `COPY` statement produces machine-readable copies of Adabas Command Log records that are supplemented with data derived by Unicenter CA-APAS. How you obtain each of these types and points you should consider are explained below.

In many cases you can override default characteristics of the output. The choices available and how to use them are discussed below.

Online Displays

Code the `INSIGHT-LINES` parameter in an `EXTRACT` or `SUMMARIZE` statement to make it possible to display current data for the request at Unicenter CA-APAS Insight terminals.

Use of the `INSIGHT-LINES` parameter has no JCL implications. `INSIGHT-LINES` causes the Data Collector to allocate additional virtual storage to hold lines awaiting display. The number of bytes of storage is approximately equal to the product of the number of lines you specify and the length of the lines. For example, if the line length is 130 bytes and you specify 100 lines, then about 13,000 bytes are required as long as the request remains active. It is a good idea to avoid making either of these factors larger than necessary.

Only the number of lines you specify in this parameter are available at any given time for viewing at terminals. In the case of an `EXTRACT` statement, the lines available are the most recent lines generated for the statement; older lines are lost as newer ones are created. In the case of a `SUMMARIZE` statement, the lines available always start at the logical beginning of the summary table.

Machine-Readable Data Files

Code the `OUTPUT-FILE` parameter in an `EXTRACT` or `SUMMARIZE` statement to produce an output file of records that contain values for specified data fields or summary functions. This parameter must specify a `ddname/filename` that has been defined in the JCL of the job in which the Data Collector is executing.

Print-Formatted Reports

Code the PRINT TO parameter in an EXTRACT or SUMMARIZE statement to produce a report formatted for printing. This parameter must specify a ddname/filename that has been defined in the JCL of the job in which the Data Collector is executing. The JCL for the specified ddname/filename determines whether the report is written directly to a print spool or is written to a regular data file intended for additional processing before printing.

Field Length and Format

Values for fields and summary functions that are specified in EXTRACT and SUMMARIZE statements are output in default lengths and formats unless you write override lengths or formats in the statements.

The default lengths and formats are shown with each field description in *Writing Requests* guide. Note that two different sets of default length and format are given for each field and summary function. One is used in print-formatted reports and online displays; this is labeled Rept Fmt. The other is used in machine-readable output files; it is labeled File Fmt.

To override the default length or format of a field or summary function in a printed report or online display, write a PF= parameter following the name of the field or summary function. To override its default length or format in a machine-readable output file, write a OF= parameter following the name. In either case, follow the detailed syntax shown in *Writing Requests* guide.

Column Headings

Printed reports and online displays are usually organized with a separate column of data for each of the specified fields or summary functions. Each column is identified by the field's or summary function's default column header unless you write an override column header for the item. The default column header is shown in each field and summary function description in the *Writing Requests* guide.

If you want to override the default column heading for a field or summary function, write a HD= parameter following its name. Follow the detailed syntax shown in *Writing Requests* guide.

Stacking Fields in a Column

You can cause a consecutive set of fields or summary functions to share a single column in a printed report or online display. For a given command or control-break entity, the value of the first field or summary function is placed in the column, then the value of the second field or summary function is stacked below that of the first, and so on.

To do this, write a slash character with a blank character on each side, for example, / , before the name of the second and each subsequent field or summary function that is to share the column.

The shared column is identified by the default column header of the first field or summary function unless you write an override column header for that item. Nothing is put into the column header for the other fields or summary functions that share the column. You will find it preferable to supply an override column header that more fully reflects the contents of the column than would the default header of the first field or summary function.

The following example shows stacking of several fields and overriding the column header.

```
SAMPLE: EXTRACT
CMD
FILE
ASSO-IO (HD='ASSO-IO-COUNT', 'DATA-IO-COUNT', 'WORK-IO-COUNT')
/ DATA-IO / WORK-IO
TOTAL-IO
DURATION
INSIGHT-LINES 50;
```

In the resulting display, the values for CMD would be in the first column, and the values of FILE would be in the second column. The values of ASSO-IO, DATA-IO and WORK-IO would be stacked one below another in the third column. TOTAL-IO would be in the fourth column and so on.

The third column's header would appear as:

```
ASSO-IO-COUNT
DATA-IO-COUNT
WORK-IO-COUNT
-----
```

Command Log Records

Use a COPY statement to obtain a machine-readable file of all or selected Adabas Command Log records. This is the only type of output COPY statements produce. For detailed information about COPY files, see the *Writing Requests* guide.

Use of COPY statements require that the named OUTPUT-FILE be defined in JCL. Coordinate with your database administrator before using a COPY statement.

Dual Files

When printed reports or output files must be accessed prior to the end of an MPM session, or the output may exceed the capacity of disk files, the Unicenter CA-APAS dual file facility may be used. For detailed information about dual files, see the *Systems Guide*. Coordinate using dual files with your database administrator.

Request Security

The ability to display or modify (PAUSE, RESUME, RESET, or DELETE) a given request can be restricted to one user or a group of users. You can activate this type of request security by adding the OWNER and SECURITY keywords to the request.

The SECURITY and OWNER keywords are not required. A request without those keywords uses the default security status. The default security status for a request is public, that is, any Unicenter CA-APAS user can display or modify the request.

OWNER Values

To assign ownership of a request, you must assign the OWNER keyword a value that matches one of the following Natural system variables for the individual or group to whom the request belongs:

- *INIT-ID
- *INIT-USER
- *APPLIC-ID
- *GROUP

SECURITY Values

The value you assign to the SECURITY keyword determines the extent to which you restrict the display or modification of an owned request. The following explains the effect of the possible values you may assign to the SECURITY keyword, from the least restrictive to the most restrictive security levels:

SECURITY=NONE

Any Unicenter CA-APAS user is allowed to display or modify the request. The same outcome is produced when the SECURITY keyword is not used in the request.

SECURITY=MODIFY

Any Unicenter CA-APAS user is allowed to display the request, but only those specified in the OWNER keyword are allowed to modify it.

SECURITY=DISPLAY

Only those users specified in the OWNER keyword are allowed to display or modify the request.

Specific Techniques for Selected Objectives

The material in this section shows how to use Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS) requests to get information about a number of different processing and performance topics. This material is only an introduction to help you get started. The flexibility and scope of Unicenter CA-APAS reporting capabilities make it possible for you to request many more kinds of information about these and other topics.

Getting Started

Default requests that come with the system are mentioned for many of the areas that are discussed. In many cases it would be useful for you to run the default request, study its output, and then decide whether you want to make an edited and renamed version of the request to more precisely meet your particular needs. We suggest that you do not delete or replace the original versions of any of the default requests.

The *Writing Requests* guide provides extensive detail about the data fields and summary functions that can be reported, as well as the options for selecting, processing, and reporting command data. Refer to that document, as needed, when creating or modifying requests.

The *Performance History System* guide describes an additional level of reporting capabilities. Information in the Performance History file, if installed at your site, has already been developed from a standard set of command data requests. You don't use command data requests to access information in the History file. Instead, you use the online inquiry functions of the History system. Additionally, you can write your own Natural programs to access the History file.

Exception Reporting of Selected Commands

A major benefit you can enjoy with Unicenter CA-APAS is the ability to let Unicenter CA-APAS monitor millions of commands to find and report those few that are of interest to you because there is something out of the ordinary about them.

For exception reporting, you usually want to focus on individual commands. Use the EXTRACT statement to report information about individual commands.

There may be occasions when you'd like to get summary information about exceptional commands, such as how many of a category of commands occur in specified time periods, how many come from the different users or are to the different files, etc. For this type of reporting use the SUMMARIZE statement.

In either case, the WHERE clause of the request is the key to defining the conditions that commands must satisfy in order to be included in a given exception report. Detailed syntax information for writing WHERE selection criteria is provided in the *Writing Requests* guide.

Your interests will probably change as you investigate a particular issue, resolve it, and determine that you no longer care to have certain categories of commands show up in one or more of your exception reports. Due to the flexibility of the Unicenter CA-APAS mechanism, you can refine the WHERE criteria of a request anytime you wish, thus increasing or reducing the size of the reports you review.

Commands Which Exceed Performance Limits

Request EXTLONG in the APSDFLT1 set of default requests in the Unicenter CA-APAS SOURCE library is an example of a request for this objective. Some possible selection criteria you might want to use are:

- Excessive I/O (TOTAL-IO, ASSO-IO, DATA-IO, WORK-IO)
- Excessive length of time in an Adabas processing thread (DURATION)
- Excessive delay before entering an Adabas thread (ENQ-TIME)
- Excessive number of descriptors updated (DESCR-UPDATED)
- Excessive CPU time (EST-CPU-TIME)

There is no one set of values for the above measures that universally marks the transition from acceptable to excessive. You need to choose limit values that are appropriate for conditions at your site. You may need to revise your limits when critical factors such as CPU model, I/O model, I/O configuration or overall Adabas workload change.

Following is an example of the EXTLONG request.

```
EXTLONG: EXTRACT(SK=1)
FILE COMMAND-CODE CMD-TYPE
ENQ-TIME (HD='ENQ TIME','CPU TIME','DURATION') / CPU / DUR
THREAD
TOTAL-IO
ASSO-IO (HD='A-IO','D-IO','W-IO') / DATA-IO / WORK-IO
DESCR-UPDATED
BUFFER-FLUSHES
ECBS
ISN (HD='ISN','ISN QUANTITY') / ISNQ
CID / CID (PF=H8)
CMDSEQ
DATE-TIME (HD='DATE - TIME','YYMMDDHHMMSS')
JOBNAME (HD='JOBNAME','NATURAL ID') / NAT-MOD-ID
TERM (HD='TERMINAL') / TERM (PF=H8)
SEARCH-BUFFER
VALUE-BUFFER
WHERE
    TOTAL-IO > 50          /* <===== adjust these limits to
    OR ASSO-IO > 40        /* <===== keep report size within
    OR WORK-IO > 10        /* <===== reason; additional
    OR DUR > 3.0 SECS      /* <===== criteria may be helpful
    OR ENQ-TIME > 1.0 SECS /* <===== use with ACBX only
TITLE = 'COMMANDS WITH EXCESSIVE TIMES OR I-O COUNTS'
PRINT TO EXTLONG;
```

Commands Which Receive Exceptional Response Codes

Request SIGRESP in APSDFLT1 and RSPCODES in the Unicenter CA-APAS Natural library are sample EXTRACT requests which report individual commands that receive specified response codes.

In addition to deciding which response codes you care to know about, there are several options you should consider when writing EXTRACT requests for this purpose.

First, you can refer to individual response code values (RESPONSE-CODE) or to classes of response codes (RESPONSE-CLASS) or to both in a given WHERE clause. For information about which response codes are included in each response class, see the appendix “RSPCLASS Definitions” in the *Writing Requests* guide. Referring to response code classes is much more concise than referring to individual response codes. You can write criteria to include or to exclude any given response codes and/or response classes.

Second, once you've decided which response codes you want reported, you have a choice of reporting all of them in a single report or of segregating them into different reports. Using a single report might sometimes result in a lengthy report that you would have to scan carefully to find particular values that might be important. Using separate requests for response codes of special importance would make it easier to find them and eliminate the possibility of overlooking them in a longer report.

If you want to know how many commands receive selected response codes, use a SUMMARIZE request such as the following example.

```
RSPSUM: SUMMARIZE
COUNT
BY RESPONSE-CODE
WHERE...
PRINT TO RSPSUM INSIGHT-LINES 50;
```

Possible variations of the BY clause include:

```
BY RESPONSE-CLASS
BY RESPONSE-CLASS BY RESPONSE-CODE
BY PROGRAM BY RESPONSE-CODE
BY NAT-MODULE-ID BY RESPONSE-CODE
BY JOBNAME BY RESPONSE-CODE
BY COMMAND-CODE BY RESPONSE-CODE
```

Commands Referencing Sensitive Files

There are several ways you could do exception reporting of accesses to sensitive files. With an EXTRACT request you can report detailed information such as user, terminal, date-time, command code, file number, ISN, format buffer and record buffer of each command that referenced the data in question. The following is an example of this type of request, reporting only attempts, successful or otherwise, to change a sensitive file. By removing or enlarging the command-code criteria, the report could reflect access activity as well as change activity.

```
WATCHBRD: EXTRACT (SK=1)
DATE-TIME
JOBNAME
PROGRAM
NAT-MODULE-ID
USER-ID
TERMINAL-ID (PF=H8)
COMMAND-CODE
RESPONSE-CODE
FILE
ISN
FB RB
WHERE FILE = 100
      AND (COMMAND-CODE = 'A1' OR = 'A4' OR = 'N1' OR = 'N2'
          OR = 'E1' OR = 'E4')
PRINT TO WATCHBRD;
```

With a SUMMARIZE request you could concisely report the volume of accesses to sensitive data, broken down by user, etc., if desired. The following example illustrates this type of request.

```
HOWMUCH: SUMMARIZE
COUNT
BY JOBNAME BY USER-ID BY COMMAND-CODE
WHERE FILE = 88 OR = 107
PRINT TO HOWMUCH;
```

Commands Whose Use is to be Restricted

You can use EXTRACT and SUMMARIZE requests to report the use of commands that are discouraged or prohibited at your site. The WHERE criteria, in addition to identifying the target commands, could also exclude any specific programs which have been authorized to use the commands. Following is an example of such a request.

```
SCRUTINY: EXTRACT (SK=1)
DATE-TIME
JOBNAME
PROGRAM
NAT-MODULE-ID
USER-ID
TERMINAL-ID (PF=H8)
COMMAND-CODE
FILE
WHERE COMMAND-CODE = 'S2' OR = 'S9'
      AND NAT-PROGRAM NOT = 'ADNU1Z'
      AND NAT-PROGRAM NOT = 'KLXINQ'
      AND JOBNAME NOT = 'PSYS12A'
PRINT TO SCRUTINY;
```

Analyzing Usage of Adabas by a Program

Use a WHERE clause in EXTRACT, SUMMARIZE and COPY requests to limit them to reporting only those Adabas commands that are issued by a particular program that you are analyzing. If multiple copies of the program are in execution, as in a production system, you may want to use additional qualifiers, such as JOBNAME or TERMINAL-ID, to restrict a request to a particular execution of the program. Following are some examples of WHERE criteria that might be useful.

```
WHERE PROGRAM = 'RPT100X'
WHERE JOBNAME = 'MYJOB01'
WHERE NAT-PROGRAM = 'JVHTEST'
WHERE NAT-PROGRAM = 'PRODXYZ' AND TERMINAL-ID = H'C1CB0075'
```

Use of an appropriate WHERE clause is assumed and is not repeated below.

Program Trace

Use an EXTRACT to generate a trace of a program's Adabas calls. The fields you list for display depend on your purpose in doing the trace. You might find it convenient to use the default request ALLTRACE as a model request from which you can generate requests tailored to your particular requirements. To do this you should:

1. Make a copy of ALLTRACE
2. Rename the copy

3. Delete any fields you do not need
4. Add a WHERE clause to target the program you want to trace
5. Use INSIGHT-LINES, PRINT TO, and/or OUTPUT-FILE according to what kind of output you want

An example of the ALLTRACE request follows.

```
ALLTRACE: EXTRACT (LS=150)
CMDSEQ (HD='SEQ')
TIME
ENQ-TIME (HD='ENQ TIME')
DURATION (HD='DURATION')
ASSO-IO (HD='ASSO')
DATA-IO (HD='DATA')
WORK-IO (HD='WORK')
FILE (HD='FIL')
COMMAND-CODE (HD='CC')
RESPONSE-CODE (HD='RSP')
CMD-TYPE (HD='T')
THREAD (HD='TH')
JOBNAME
TERMINAL-ID (HD='TERM')
TERMINAL-ID (HD='TERM-HEX' PF=H8)
NAT-MOD-ID (HD='Natural MODULE ID')
COMMAND-ID (PF=H8)
ISN-QUANTITY (HD='ISQ')
INSIGHT-LINES = 50;
```

Program Profile

When analyzing the Adabas performance of a program, it's often quite informative to produce a summary profile of the calls from one or more executions of the program. Such a profile could be at various levels of summarization. A simple approach that's generally useful for getting a good understanding of a program's Adabas activity is shown in the following example.

```
PROFILE: SUMMARIZE
COUNT
MEAN(ASSO-IO) MAX(ASSO-IO)
MEAN(DATA-IO) MAX(DATA-IO)
MEAN(WORK-IO) MAX(WORK-IO)
MAX(HOLD-COUNT)
BY FILE BY COMMAND-CODE
WHERE...
INSIGHT-LINES 30;
```


Analyzing Search Logic

If a trace or profile of a program shows that search commands from the program are causing excessive numbers of I/Os to the Associator or Work, then you might find it useful to get more information about the search commands. One item of interest is the number of records being qualified by each search command. Display ISN-QUANTITY in EXTRACT traces or MEAN(ISN-QUANTITY) and MAX(ISN-QUANTITY) in SUMMARIZE profiles such as those described previously.

Excessive ISN quantities could indicate that the search criteria being used are not adequate to define the set of records wanted. Reasonably small ISN quantities could indicate that one or more of multiple search criteria is causing excessive intermediate ISN lists to be written to and read from work; such a criterion could perhaps be eliminated or replaced with something more selective.

Other items of interest are the names of the descriptors, the operators, and the values being referenced in each search. Display FORMAT-BUFFER and VALUE-BUFFER in EXTRACT traces to get this information.

Range searches (field = value1 THRU value2) where the range is from minimum absolute value through maximum absolute value are totally useless for search purposes and can cause considerable unnecessary I/O to both Associator and Work.

In some cases it may be useful to tie sets of search commands back to a single Natural FIND statement. Display COMMAND-ID in hexadecimal format and NAT-STMT-NO in EXTRACT traces. Add NAT-STMT-NO as an intermediate BY field in SUMMARIZE profiles as shown below.

BY FILE BY NAT-STMT-NO BY COMMAND-CODE

Investigating Hold Queue Overflow

When the Adabas Hold Queue is full, online users of Adabas can be delayed. Unless the Adabas operator command, STOPU, is used to delete one or more users in order to free up space in the Hold Queue, the delays may last as long as the value of the ADARUN transaction time limit parameter, TT. Since TT is typically set for three to five minutes, even 10 to 20 minutes at some sites, the delays can be serious. There are basically two causes of Hold Queue overflow:

- The size of the Hold Queue is too small for the current mix of users, even though all of the application programs in use have appropriate logic for holding and releasing records
- One or more currently active programs is holding an unreasonably large number of records

If Hold Queue overflow is suspected as the cause of processing delays, you can use Unicenter CA-APAS to diagnose the situation. With Unicenter CA-APAS you can display the total number of records currently being held, the highest number of records being held by any one user, and the identification of that user.

The SNAP command causes the items mentioned above to be displayed at the end of each interval that is in effect for the SNAPSHOT request. If you want the information displayed more frequently, then DISPLAY the HOLDQ request shown below; this can be done as often as you wish to press the ENTER key.

```
HOLDQ: SUMMARIZE (SF=3)
COUNT
NO-USERS
HELD-RECORDS
MAX-HELD-RECORDS
MAX-HOLDER-JOB
MAX-HOLDER-ID
MAX-HOLDER-ID (PF=H8)
INTERVAL 0 SECONDS
INSIGHT-LINES 5;
```

If the highest number of holds by any one user is reasonable, then the Hold Queue should be larger in future MPM sessions. On the other hand, if a user holds too many records, then you can identify the user to delete and the program to correct.

With Natural security, there is an important operational consideration regarding the use of Unicenter CA-APAS to analyze Hold Queue overflow situations. While the Hold Queue is full, you can't start a Natural session or LOGON to a different Natural application, because Natural Security must hold records to carry out either of these operations. Therefore, your terminal must already be logged on to Unicenter CA-APAS before the Hold Queue overflow condition begins.

Analyzing Hold Logic

The field HOLD-COUNT can be helpful in diagnosing problems with the hold logic of programs. In an EXTRACT request, display HOLD-COUNT. In a SUMMARIZE request, display MAX(HOLD-COUNT), such as in the following request:

```
XCESHOLD: SUMMARIZE (SUBTOT=OFF)
MAX(HOLD-COUNT) SORT DESC
BY NAT-PROG
WHERE HOLD-COUNT GT 50
INSIGHT-LINES 50;
```

If a program loops reading the same ISN with hold, then the value of HOLD-COUNT is exaggerated, sometimes even overflowing its accumulator and causing asterisks to be output instead of numeric values. Displaying field ISN in an EXTRACT can confirm this cause of unreasonable or overflow values; the value of the ISN that is being reread may help in debugging the program.

Reasonable upper limits for the number of records held at any point vary from application to application. A general rule of thumb is to investigate any cases where more than 50 records are held. Where hundreds of records are held, program logic is usually faulty. Possible problems are:

- Unintended program looping
- Reading all records with hold where only a small percentage of the records are ultimately updated or deleted
- End transaction (ET) commands not being issued frequently enough

Analyzing Peak Processing Times

Use a SUMMARIZE request to develop profiles of the overall magnitude of the Adabas workloads and Adabas processing efficiency across successive time periods. Some factors that determine how useful the request outputs are to you include:

- The size of the time periods reported
- Your choice between BY and INTERVAL for specifying reporting periods
- The fields you report

Each of these factors is discussed below.

Sizes of Time Periods

Sizes of time periods useful for analyzing peak processing periods could range from a maximum of one hour to a minimum of five minutes or so. Periods of fifteen minutes seem to be a good general compromise in the tradeoff between fineness of data and number of periods to study.

Choosing Between BY and INTERVAL

If the output is for manual review, whether printed reports or online displays, it is easier to compare successive time periods if the periods are specified in a BY clause. This allows multiple successive periods to be seen and compared on each page or screen. This approach requires the size of INTERVAL to be either HOUR or QUARTER-HOUR.

The Fields You Report

The following fields are suggested for depicting the overall workload and Adabas processing efficiency of each reported time period:

- COUNT and CMD-RATE - gross command load; quantity and rate
- SUM(TOTAL-IO) and RATE(TOTAL-IO) - gross I/O load; quantity and rate
- MEAN(ENQ-TIME) and MAX(ENQ-TIME) - indicators of Adabas backlog
- MEAN(ECBS) and MAX(ECBS) - indicators of Adabas backlog

To reflect overall performance, do not use a WHERE clause

To analyze update activity, use two requests, one selecting commands with COMMAND-TYPE = 'U' and the other with COMMAND-TYPE NOT = 'U'. Default request UPDSAT in the Unicenter CA-APAS library is an example of a request that reports update processing across time periods. In an ADASMP environment, select SMPID=1 to isolate update commands.

Determining Major Sources of Workloads

To find out the major sources of Adabas processing workloads, use a SUMMARIZE request with the SORT DESCENDING option. Key factors to consider in writing the request include:

- Choice of entity to report BY
- Choice of summary function to sort on
- Choice of Entity

Typical entities to tie workloads back to (using the BY clause) are:

- NAT-PROGRAM
- PROGRAM
- JOBNAME
- TERMINAL-ID
- USER-ID

Choice of Summary Function

To rank users by number of commands issued, write SORT DESCENDING after the summary function COUNT. To rank users by number of I/Os their commands caused, write SORT DESCENDING after the summary function SUM(TOTAL-IO). To measure overall workload, do not use a WHERE clause. To measure only certain types of workload (update, non-update, batch, TP, selected files, etc.), use a WHERE clause to select only those commands that are of interest. To report the workload broken down into components, add BY fields before or after those mentioned above. Some examples of this approach are shown below.

```
BY FILE BY PROGRAM...  
BY USER-TYPE BY NAT-PROGRAM...  
BY TERMINAL-ID BY NAT-PROGRAM...
```

Detecting Adabas Processing Bottlenecks

Ways to measure potential bottlenecks in Adabas processing are discussed below.

Update Commands

One method to learn the degree to which update commands must wait for a thread is to compare the mean and maximum values of ENQ-TIME for update commands across successive time periods of about five to fifteen minutes. Default request UPDSAT in the Unicenter CA-APAS Natural library is an example of a request that can be used for this purpose.

Another approach is to compare the mean and maximum values of ENQ-TIME for update commands with those for simple commands. The more these values for update commands exceed those for simple commands, the greater the impact of this bottleneck. Default request ENQUEUE in the Unicenter CA-APAS Natural library illustrates this approach.

Complex Searches

Default request ENQUEUE in the Unicenter CA-APAS Natural library can be used to detect delays in starting “complex” searches. The more that the MEAN and MAX values of ENQ-TIME for complex commands exceed those for simple commands, the greater the impact of this kind of bottleneck.

```
ENQUEUE: SUMMARIZE (SF=3)
MIN(TIME) /* approximate interval start time
MAX(TIME) /* approximate interval end time
COUNT MEAN(ENQ-TIME)
MAX(ENQ-TIME)
BY QUARTER-HOUR COMMAND-TYPE
PRINT TO GENSUM /* enable for printed report
INSIGHT-LINES = 60;
```

Shortage of Threads

To detect a shortage of non-update threads, monitor the count of commands being processed in the highest numbered thread. Refer to any of the following:

- Session statistics printed by the Adabas nucleus
- Output from DSTAT operator commands
- A request that reports commands by thread

A command count of zero for the highest thread indicates the absence of this kind of bottleneck. The greater the count, the more times commands might have had to wait for an available thread.

Default request THREADS in the Unicenter CA-APAS library can be used for this purpose. If this is the only reason for using this request, add a WHERE to select only the highest numbered thread; this will reduce the overhead for the request to close to nothing.

```
THREADS: SUMMARIZE  
COUNT  
PERCENT  
PCT(TOT-IO)  
PCT(DURATION)  
MEAN(TOT-IO)  
MEAN(DURATION)  
MEAN(ENQ-TIME)  
MEAN(ECBS)  
MAX(ECBS)  
BY THREAD BY COMMAND-CODE  
INSIGHT-LINES = 60;
```

Analyzing Impact of Batch Jobs on Response Times

Default request THRDUSRS in the Unicenter CA-APAS Natural library allows the comparison of loads from batch jobs and online users in each thread, for successive time periods. You could refine this request to show percentages, use graph reporting format, etc.

To measure the impact of batch jobs on the response times of online users, look for the following indications.

For update processing, examine the statistics for thread 1. See whether the mean ENQ-TIME for commands from TP users increases as the number of commands from batch users increases. This increase in TP ENQ-TIME is one component of degraded response times.

For non-update processing, see if batch commands start to dominate in lower threads causing TP commands to be effectively shifted to higher threads than when no batch jobs are running. In a CPU-bound environment, higher threads result in larger values of DURATION as thread number increases. The increase in duration is another component of degraded response times.

```
THRDUSRS: SUMMARIZE (SUBTOT=OFF)
MIN(DATE) MIN(TIME)
COUNT
SUM(DUR)
MEAN(ENQ-TIME)
BY QUARTER-HOUR
BY THREAD
BY USER-TYPE
INSIGHT-LINES=60;
```

Determining CPU-Bound versus I/O-Bound

Use default request THRD CMD S in the Unicenter CA-APAS Natural library to get an idea as to whether Adabas processing tends to be CPU bound or I/O bound. Refer to the comments in the request for interpretation of the reported values.

Identifying Candidates for Prefetch/Multifetch

Use default request PREFLSB in the Unicenter CA-APAS Natural library to identify batch jobs whose execution time could be improved by using the Adabas prefetch option. This request is a summary of the Lx type commands generated by batch jobs.

Likely candidates are reported in descending order of potential for improvement, based on the number of sequential read commands issued. Jobs that contain only one Lx sequence with few or no other command types are the optimum candidates. Candidates should then have their code reviewed to determine whether prefetch should be invoked.

For each candidate, the request reports data that helps in setting the prefetch parameter, PREFLSB. PREFLSB should be set somewhere between the MEAN and MAX values of REC-LENGTH-DEC. When prefetch is specified for a job via ADARUN parameters, Adabas makes the following calculation:

$$X = (\text{PREFLSB} - (128 + \text{FBL})) / \text{RBL}$$

If X is less than eight, prefetch is not activated. Use the MEAN values of FBL and RBL from the request to determine the outcome of this test. If X is less than 8, adjust the PREFLSB value accordingly to insure that prefetch is used.

Evaluating Effectiveness of ADAM

To measure the effectiveness of using ADAM files, measure the MEAN values of ASSO-IO and DATA-IO for search commands that should have used the ADAM technique to find and read a record. A sample request is shown below.

```
ADAMEFF: SUMMARIZE
COUNT
MEAN(ASSO-IO)
MEAN(DATA-IO)
BY FILE
WHERE FILE = nnn OR = mmm... /* select only ADAM files
                                AND ASSO-IO LE 4 /* exclude non-ADAM searches
                                AND WORK-IO = 0 /* exclude buffer flushes
                                AND DATA-IO LE 2 /* ditto
                                AND RESPONSE-CODE = 0
                                AND COMMAND-CODE = 'S1' OR = 'S4'
                                AND FBL GT 1
....
```

The WHERE criteria shown above are intended to limit the report as much as possible to those successful S1 and S4 commands which read the first qualifying record and are to ADAM files, excluding any commands which happened to trigger a buffer flush, or any complex searches. The criteria are only approximate in this respect.

Optimum ADAM efficiency would be reflected by MEAN(ASSO-IO) values of zero and MEAN(DATA-IO) values of 1.0 or less. As either of these values goes up, the ADAM efficiency is less.

Optimizing Logical Sequential Processing

Use default request L3DEUSE in the Unicenter CA-APAS Natural library to find out whether the records of a given file are physically sequenced in Data Storage to give the lowest MEAN(DATA-IO) for the descriptor that is most often used for logical sequential reading.

Using Unicenter CA-APAS Online

This chapter describes the functional capabilities, operational procedures, and suggestions for effective use of Insight, the interactive real-time monitor component of Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS).

It is intended for those who use Unicenter CA-APAS from terminals to analyze the current processing of one or more active Adabas sessions. Database administrators, system programmers, and application programmers or analysts concerned with Adabas performance are typical members of this audience.

It is also useful to those who are evaluating the potential of Unicenter CA-APAS to help respond promptly and effectively to Adabas performance problems while the problems are being experienced rather than after the fact.

Overview

Unicenter CA-APAS has an interactive component meant to provide real-time monitoring of Adabas activity. The major benefit of using Unicenter CA-APAS is the ability to deal with Adabas processing while it is happening. Real-time, interactive reporting of Adabas performance information enables:

- More effective response to complaints of poor online response or excessive batch job run times
- More timely, convenient, and economical monitoring of Adabas calls from specific programs
- More informed analysis of current overall activity of a copy of the Adabas nucleus, with more accurate isolation of specific causes of performance problems

These advantages of Unicenter CA-APAS stem largely from the ability to quickly create or modify, activate, and display data from Unicenter CA-APAS requests tailored to unique circumstances of the moment.

Unicenter CA-APAS Capabilities

You can invoke Unicenter CA-APAS from Natural to obtain information about the processing being done by one or more active copies of the Adabas nucleus. All of the reporting capabilities of Unicenter CA-APAS, as described in the *Writing Requests* guide, are available through the Unicenter CA-APAS Natural user interface, Insight. Two sets of programs must function in order to obtain such information:

1. The Data Collector must be executing with each copy of Adabas that is to be monitored. The Data Collector must be invoked as part of the Adabas start-up procedure; Unicenter CA-APAS terminal users do not start the Data Collector.
2. Unicenter CA-APAS Insight terminal interface programs are executed during an online Natural session. These programs communicate with the Data Collector to manipulate Unicenter CA-APAS requests and have request outputs sent to your terminal for display.

The Unicenter CA-APAS Insight terminal interface communicates with some of the facilities of Natural and with one or more copies of the Data Collector to control Unicenter CA-APAS requests and display information from active requests. You may:

- Use standard Natural source code facilities to create, modify, save and retrieve Unicenter CA-APAS request source code
- Direct Data Collectors to syntax check, add, pause, reset, resume and delete requests
- Display current request data at your terminal in single or dual-window screen formats with horizontal and vertical scrolling options
- Cause Data Collectors to switch the open/close status of Unicenter CA-APAS dual output files
- Dynamically modify selection criteria and buffer output options for standard command logging by Adabas
- Dynamically modify command selection criteria and Adabas buffer output options for command logging to Unicenter CA-APAS COPY files

Conducting a Unicenter CA-APAS Insight Session

The following material describes how to use Unicenter CA-APAS Insight from your terminal. Similar information is available from the Unicenter CA-APAS HELP facility.

Basic Procedures

There are just a few basic procedures to learn; ease of use allows you to focus on the Adabas performance information that is the true objective of a Unicenter CA-APAS Insight session.

A Unicenter CA-APAS Insight session is initiated from within a Natural session by the Natural command sequence:

```
LOGON INSIGHT
INSIGHT
```

The logon id may be different at your site. Check with your database administrator.

Initial Screen

The first screen of a Unicenter CA-APAS Insight session, shown below, identifies the version and system maintenance (SM) levels of the products involved.

```

      ADABAS PERFORMANCE ANALYSIS SYSTEM (CA-APAS)
    COPYRIGHT 2003 COMPUTER ASSOCIATES INTERNATIONAL, INC.
  ALL RIGHTS RESERVED. CONTAINS TRADE SECRETS AND CONFIDENTIAL PROPRIETARY
    INFORMATION OF COMPUTER ASSOCIATES INTERNATIONAL, INC.
  REVERSE ENGINEERING PROHIBITED. COPYRIGHT NOTICE DOES NOT IMPLY PUBLICATION.
```

```

IIII   NNN   NN   SSSSSS   IIII   GGGGGGGG   HH   HH   TTTTTTTT
II     NNNN  NN   SS       II     GG         HH   HH   TT
II     NN NN  NN   SS       II     GG         HH   HH   TT
II     NN NN  NN   SSSSSS   II     GG         HHHHHHHH  TT
II     NN   NN NN   SS       II     GG   GGGG   HH   HH   TT
II     NN   NNNN   SS       II     GG   GG     HH   HH   TT
IIII   NN     NNN   SSSSSS   IIII   GGGGGGGG   HH   HH   TT
```

```
-----
I N T E R A C T I V E   A D A B A S   P E R F O R M A N C E   M O N I T O R
-----
```

CA-INSIGHT V4.01.01 FOR ADABAS V7 & NATURAL V3

```

CMD: _____ REQ: _____ DBID: ____0 SMPID: ____0 > USE PF KEY, CMD, OR ENTER <
PFK: 1=HELP 3=MENU 4=SELECT 5=SHOW 6=DISPLAY 9=SNAP 12=REQMGT INSTART
```

This screen also provides options for invoking the first Unicenter CA-APAS Insight functions:

- Press the ENTER key to proceed by default to the Main Menu screen
- Press any PF-key shown on the bottom line of the screen to invoke its associated function
- Type any valid Unicenter CA-APAS Insight command into the CMD field (also using the REQ, DBID and SMPID fields as necessary) on the bottom line of the screen, then press the ENTER key

The bottom two lines of most Unicenter CA-APAS Insight screens have the same fields for commands and similar reminders for PF-key options. The top line of all screens is used for confirmation or error messages.

Entering Unicenter CA-APAS Insight Commands and Operand

The various ways of invoking Unicenter CA-APAS Insight functions are:

- By entering a selection code on a MENU screen, then pressing the ENTER key
- By pressing a designated PF-key
- By typing a valid command keyword or abbreviation into the CMD field, then pressing the ENTER key

The Unicenter CA-APAS Insight Main Menu is shown below:

```

C A - I N S I G H T      M A I N      M E N U

CODE  FUNCTION                                           (COMMAND)
-----
1     INFORMATION ABOUT CURRENT CA-APAS REQUESTS      (SHOW)
2     DISPLAY DATA FROM CURRENT REQUEST(S)           (DISPLAY)
3     SELECT A SAVED REQUEST TO START OR EDIT         (SELECT)
4     REQUEST MANAGEMENT MENU

5     TRACE      - (GENERATE AND START NEW REQUEST)
6     SUMMARY    -      ''
7     EXCEPTION  -      ''

8     ADABAS NUCLEUS PROCESSING STATISTICS            (SNAP)
9     ADABAS FUNCTIONS MENU                            (ADAMENU)
H     HELP ABOUT CA-INSIGHT
.     TERMINATE CA-INSIGHT
A     CA-APAS HISTORY APPLICATION
S     CA-SPACEMAN APPLICATION

CODE:  _      USE MENU CODE, PF-KEY, OR CA-INSIGHT COMMAND

CMD:  _____ REQ:  _____ DBID:  _____ 0 SMPID:  _____ 0 DT: 02-11-20 TM: 19:41:15.7
PFK:  1=HELP  3=END  4=SELECT  5=SHOW  6=DISPLAY  9=SNAP  12=REQMGT  INMENU

```

Some commands require entry of a Unicenter CA-APAS request name in the REQ field; the REQ field may also be used to enter a JCL ddname/filename with a FLIPXY command to cause the Data Collector to reverse the opened/closed status of a pair of dual output files.

The active Adabas nucleus to be referenced may be specified by entering a database-id value in the DBID field. If the DBID field value is left at zero, Unicenter CA-APAS Insight uses a default database-id set by your site.

Entering Natural Commands and Operands

Natural's LIST, READ, EDIT, and SAVE commands may also be entered in the CMD field; valid operands may be entered in the REQ field. It is assumed that you are already familiar with these commands.

The main use of these Natural functions during a Unicenter CA-APAS Insight session is preparing Unicenter CA-APAS requests for submission to the Data Collector; this is discussed in more detail later. They may also be used to create and execute Natural programs whose Adabas calls you wish to monitor.

Using Unicenter CA-APAS Insight HELP

The Unicenter CA-APAS Insight HELP facility may be invoked by command or by PF key. Much of the information in this section is available through help selections. Invoking help normally results in the display of the HELP Main Menu from which major topic areas may be selected.

To simplify getting information about selected errors, the HELP Main Menu and the Error Selection Sub-menu are bypassed if an error code is present when HELP is invoked. In such cases, the first HELP screen addresses the specific error. You may then leave HELP or select any of the other help choices.

Continuing an Interrupted Unicenter CA-APAS Insight Session

Various events may cause the currently executing Unicenter CA-APAS Insight module to be interrupted with a resulting NEXT prompt from Natural. The Unicenter CA-APAS Insight session may usually be resumed by simply entering the command:

next INSIGHT

If this is not successful, terminate the Unicenter CA-APAS Insight session as discussed below and start a new Unicenter CA-APAS Insight session. If still unsuccessful, terminate the Natural session and begin a new Natural session.

Terminating a Unicenter CA-APAS Insight Session

Your Unicenter CA-APAS Insight session continues until one of the following occurs:

- END, FIN, QUIT or "." is typed in the CMD field and the ENTER key is pressed
- PF3 is pressed from the Unicenter CA-APAS Insight Main Menu panel
- The Natural session ends
- Unicenter CA-APAS global variables are released. Note that a Natural LOGON command releases global variables

Note that FIN terminates the Natural session as well as the Unicenter CA-APAS Insight application session.

Proper termination of a Unicenter CA-APAS Insight terminal session is important if the Natural session is to continue for other work. Use of the termination options provided by Unicenter CA-APAS Insight assures that all necessary housekeeping functions are performed.

In situations where Unicenter CA-APAS Insight has been exited by other than the standard exit options, you may force proper termination by entering:

next INQUIT

Continued execution of the Data Collector within Adabas sessions is not affected by termination of Unicenter CA-APAS Insight terminal sessions. Any Unicenter CA-APAS requests started by Unicenter CA-APAS Insight terminal users and not deleted continue to be processed by the Data Collector in the affected Adabas region.

Manipulating Unicenter CA-APAS Requests

Information that you want about the processing in an active Adabas region must be specified in Unicenter CA-APAS requests. For information describing request syntax and fields that may be displayed, see the *Writing Requests* guide.

To cause Adabas processing information to be generated, a request must be submitted to the Data Collector in a given Adabas region for syntax checking and data gathering. This may be done by including one or more requests as Unicenter CA-APAS input when starting an Adabas nucleus session and/or by submitting requests to the Data Collector via the Unicenter CA-APAS Insight terminal interface.

The phases of manipulating Unicenter CA-APAS requests with the Unicenter CA-APAS Insight terminal interface are described below. Each execution of Unicenter CA-APAS Insight commands discussed below applies to only one copy of the Adabas nucleus. If a request is to be used with multiple copies of the nucleus, commands need to be executed for each copy. The DBID and SMPID fields are used to specify a particular Adabas nucleus session.

Automatic Generation and Activation of New Requests

The easiest way to create and start a new Unicenter CA-APAS request is by choosing Trace, Summary or Exception from the Main Menu. No knowledge of Unicenter CA-APAS request syntax or Unicenter CA-APAS commands is required. Self-guiding prompt screens allow you to mark data fields to be included and command selection criteria. The new request is generated and started, then the DISPLAY function is automatically invoked for the new request.

This approach to request creation is primarily intended for new or occasional users of Unicenter CA-APAS. It does not attempt to offer the full range of Unicenter CA-APAS request capabilities. To take advantage of more request capabilities, use the capabilities described next.

Selection of Saved Requests

A large collection of predefined or default Unicenter CA-APAS requests are provided in the Unicenter CA-APAS Insight Natural library as Natural source programs. Locally developed requests may also be SAVED into the library.

Unicenter CA-APAS Insight SELECT function, shown below, displays a list of the names and first lines of the saved requests.

```

      R E Q U E S T S   I N   C U R R E N T   L I B R A R Y

ACBXMISS * COMMAND LOGS WITHOUT ACBX INFORMATION
ACBXVRFY * USED TO VERIFY ACBX INSTALLATION UNDER ANY ADABAS VERSION
ADARUN   * SPECIAL REQUEST WHICH MUST BE ACTIVE FOR "ADARUN" COMMAND
ALLCMDS  * OUTPUT FILE SUMMARY OF ALL COMMANDS FOR QUARTER HOUR INTERVALS
ALLTRACE * LIST ALL INDIVIDUAL COMMANDS
BFLADA   * SUMMARY OF I/O BUFFER FLUSH COUNTS FROM ADABAS
BFLUSHES * STATISTICS FOR ASSUMED ADABAS I/O BUFFER FLUSHES, BY CMD CODE
BFLX     * "DECLARE" A BUFFER-FLUSH INDICATOR, BFLX
BIGFIND  * IDENTIFY SEARCH COMMANDS QUALIFYING EXCESSIVE NUMBERS OF RECORDS
BYMINUTE * GRAPHIC DISPLAY OF COMMAND COUNTS BY MINUTE OF DAY
CASOXXX  * SPACEMAN ONLY - MOST READ ASSO RABNS FOR ADARUN CASO... PARMS
CDATAXXX * SPACEMAN ONLY - MOST READ DATA RABNS FOR ADARUN CDATA... PARMS
CLOSTIME * REPORT TOTAL RESOURCE USAGE FROM THE 'CL' COMMAND
CLXTRACE * LIST ALL INDIVIDUAL "COMPLEX" SEARCH COMMANDS
CMDSUM   * PERCENT OF CMDS & I/O BY COMMAND CODE WITHIN FILE NUMBER
CTYPLOAD * SUMMARY OF LOADS BY COMMAND-CODE WITHIN COMMAND-TYPE
CWORKFAC * SPACEMAN ONLY - MOST READ WORK RABNS FOR ADARUN CWORK.FAC PARMS
DATALOC  * MAY REFLECT RELATIVE DATA I/O SERVICE TIMES BETWEEN FILES
MARK 1 REQUEST: E-EDIT S-START -OR- TO RESTART BROWSE, PUT START VALUE IN REQ:
CMD: _____ REQ: _____ DBID: _____ SMPID: _____ DT: 02-11-20 TM: 19:43:57.7
PFK: 1=HELP 3=MENU 5=SHOW 6=DISPLAY 9=SNAP 12=REQMGT INSELECT

```

Pressing the ENTER key causes the browse of saved requests to continue. A new starting value may first be entered in the REQ field to re-position the browse to another point in the alphabet. One request from any screen may be selected by its number and marked for starting or for editing.

Starting Saved Requests

If you already know the name of a saved request that you want to start, you can simply:

1. Type START in the CMD field.
2. Type the request name in the REQ field.
3. Press the ENTER key.

Preparing Unicenter CA-APAS Requests for Processing

Standard Natural source code facilities may be used to create, edit, save, and retrieve requests. To simplify using Natural facilities, Unicenter CA-APAS Insight accepts the Natural commands EDIT, LIST, READ, and SAVE in the CMD screen entry field and invoke them via Natural's STACK facility. Unicenter CA-APAS Insight resumes execution upon completion of the Natural function.

A properly coded request must be present in the Natural source code work area before it can be passed to the Data Collector. You may use Natural's READ or EDIT commands to bring existing requests into the source work area from the Natural source library, or you may use the Natural full-screen editor to key new requests into the source work area.

Each request should be saved under a name identical to the label in the request source. The first line of each request should be a comment that describes the request; this line is displayed by the Unicenter CA-APAS Insight SELECT function when users are browsing the saved requests.

Request syntax is described in the *Writing Requests* guide.

Syntax Checking and Activating Requests

Once present in the Natural source work area, a request may be passed to the Data Collector by either a CHECK or ADD command. Both CHECK and ADD cause the request to be syntax checked; ADD also causes data generation to begin for the request if syntax checking is successful.

If the CHECK or ADD command results in syntax errors, a listing of the request text and syntax checker messages is displayed. If the length overflows the screen height you may scroll vertically through the listing. Scrolling is discussed later.

Resetting Request Data

The RESET command causes the data accumulators of an active SUMMARIZE request to be output to its PRINT TO and/or OUTPUT-FILE and then set to zeroes. Data gathering for the request resumes from the zero values.

RESET causes the INSIGHT-LINES buffer of an EXTRACT request to be set to blanks, thus erasing the data and generating a blank line as a break between display lines generated before the RESET and those generated after the RESET.

RESET may also be invoked from the SHOW screen, which is described later.

Suspending and Restarting Requests

An active request may be suspended with the PAUSE command and later restarted with the RESUME command. Adabas processing between the PAUSE and the RESUME is excluded from the request's data. Suspending a request eliminates the CPU overhead of the request in the Adabas region, but does not eliminate its virtual storage overhead.

PAUSE and RESUME may also be invoked from the SHOW screen that is described below.

Purging Requests from the Data Collector

The DELETE command purges a named request from the Data Collector. Deleting a request eliminates all of its storage and CPU overhead from the Adabas region.

DELETE may also be invoked from the SHOW screen that is described below under the heading "Showing Request Status".

Sets of requests can be deleted using the following commands:

Command	Description
DELBYNAM	Deletes all requests with names beginning with the prefix given in the 'REQ:' field. A trailing asterisk, *, is allowed, but not required.
DELBYOWN	Deletes all requests with owners specified by the value given in the 'REQ:' field, as follows: '*' all public requests 'xx...xx' exact match on full owner value 'xx...xx*' owners beginning with the prefix before trailing '*'

Showing Request Status

The SHOW command displays the name and status of all requests currently known to the Data Collector in a given Adabas session, whether they were submitted as part of the Adabas job or by Unicenter CA-APAS Insight terminal users. A detailed explanation of the data items on the SHOW display is available through HELP.

```

STATUS OF APAS REQUESTS FOR DBID:      0
  REQ NAME  TY  INTVL  ST  INLN  TSIZE  PRINT TO  OUT-FILE  OWNER  1 OF 31 SEC
-----
  HIWATER   S           A   0016  00001
  SNAPSHOT  S   00000  A   0022  00001
  L3DEUSE   S           A   0030  00005
  PREFLSB   S           A   0020  00004
  THREADS   S           A   0060  00009
  THRDUSRS  S           A   0060  00011
  THRDCMDS  S           A   0060  00007
  ACBXVRFY  D           A   0200
  UPDSAT    S           A           00003  GENSUM
  MAXIUBL   S           A   0005  00001  GENSUM
  SRCHDES   S           A           00003  GENSUM
  ADAMEFF   S           A           00012  GENSUM
  RECLEND   S           A           00003  GENSUM
  RECLENC   S           A           00003  GENSUM
  L3SEQ     S           A           00005  GENSUM
  BFLUSHES  S           A           00001  GENSUM
  TIOSUM    S           A           00007  GENSUM
MARK 1 REQUEST: D-DISPLAY -OR- ANY REQ(S): R-RESET P-PAUSE U-RESUME X-DELETE
CMD:      REQ:      DBID:      0 SMPID: 0 DT: 02-11-20 TM: 19:49:52.8
PFK: 1=HELP 3=MENU 4=SELECT 6=DISPLAY 9=SNAP 12=REQMGT INSHOW

```

If the list of current requests on the SHOW screen exceeds the screen height, simply press the ENTER key to advance through the list.

Predefined Requests

A number of requests are provided with the Unicenter CA-APAS system. They are stored as Natural source programs under the application-id, INSIGHT. They may be used as provided or as models from which customized requests may be derived.

The predefined requests should be retained exactly as provided; modified versions should be labeled and saved under different names. Note that the ADARUN, SNAPSHOT, and HIWATER requests are special cases. There should be only one request by each name. The text of these requests should never be modified.

Displaying Request Data

The DISPLAY function presents current data from active requests.

Standard Functioning of the DISPLAY Command

While a request is active and the Data Collector in the Adabas region is gathering data for it, a DISPLAY command causes your Unicenter CA-APAS Insight session to obtain the latest data for the request. The first data lines are displayed.

The data lines may exceed the width or height of your screen. After the first window of data has appeared on the screen as a result of the DISPLAY command, you may scroll the set of returned lines within the screen. Mechanics of scrolling are described later.

Scrolling precludes execution of any other command at the same time, even if a non-scrolling command value is present in the CMD field. Therefore, scrolling only repositions within the set of data lines previously acquired by the DISPLAY command; scrolling never causes the acquisition of new data lines from the Data Collector. Another DISPLAY command, which precludes scrolling at the same time, must be executed in order to obtain more recent data for the request.

Special DISPLAY Option

The DISPLAY function is used frequently; to reduce the keystrokes required, a blank value in the CMD screen entry field is treated as a synonym for DISPLAY.

Scrolling Data

The CHECK, ADD and DISPLAY commands cause lines of information to be sent from the Data Collector to the Insight terminal interface. The set of lines returned for one of these commands may exceed the size of the screen. You may use the scrolling facility described below to reposition a set of data lines within the screen.

Scrolling Procedure

Before scrolling is possible, a command must have returned a set of lines to the screen. Then any number of scrolling operations may be executed before a new set of lines is obtained for that screen from the Data Collector.

To scroll, you may optionally enter a valid scroll amount value into the SCROLL screen entry field located on the bottom line of the screen. Valid amounts are:

- FULL, or the letter F, to shift by a full window width or height
- HALF, or the letter H, to shift by half a window width or height
- A number of columns or lines to shift

FULL is the default amount if no value is entered.

Next, you must specify a direction for “moving” the screen display by one of the following methods:

- Press the PF key shown on the screen for the desired direction. Alternate PF keys that may be used are PF19-up, PF20-down, PF22-left, and PF23-right.
- Enter a scrolling direction command: DOWN or the letter D, UP or the letter U, LEFT or the letter L, or RIGHT or the letter R in the CMD field and press the ENTER key.

Logic and Considerations

If the line width or number of lines available exceeds the requested scrolling amount, the amount by which you move vertically or horizontally through the report is less than what you request. The message “SCROLL LIMIT REACHED” is issued if scrolling is attempted when there is no additional data in the specified direction.

No other functions may be executed along with a scrolling operation; any other type of command value present in the CMD screen entry field is ignored until some action other than scrolling is specified.

As long as consecutive scrolling operations are executed, the only lines available for viewing are the ones returned prior to the beginning of scrolling actions. To obtain more recent lines, an appropriate non-scrolling command must be executed. The status line in any window is never scrolled.

ADD and CHECK commands return lines that contain request text and syntax messages. All of these lines are subject to vertical scrolling.

DISPLAY commands return lines that stem from a request definition and execution. Title lines are exempt from scrolling. Column header lines are exempt from vertical scrolling, but they are included in horizontal scrolling. Data lines may be scrolled in any direction.

SNAP Display of Adabas Nucleus Activity

The predefined request named SNAPSHOT is handled differently in some respects from other requests. The differences are:

- Data from this request is obtained by executing a SNAP command, not a DISPLAY command
- The REQ field is ignored when SNAP is processed
- Data from the SNAPSHOT request is reformatted by Insight and is displayed in a special full-screen format
- Since all of the data from a given execution of the SNAP command fits onto one screen, scrolling is not applicable to this request

The SNAPSHOT request contains a time interval specification. The data values returned by SNAP commands are changed only when an interval has completed. Multiple SNAP commands executed within a given interval all return the same data values. You may wish to make the interval value smaller or larger. An interval value of zero causes each SNAP command to display data for all processing since the previous SNAP command.

Switching Dual Output Files

The Data Collector in an Adabas region writes records to the following kinds of output files if so directed by requests:

- Print files
- Data files
- COPY files

Unicenter CA-APAS provides the option of using dual physical files for any of these types of logical output file.

When dual physical files are being used for a logical output file, records are written to one of the physical files until it is full or until a user intervenes, as described below. Then that output file is closed, the other is opened; further output is written to the newly-opened physical file.

A switch between the pair of physical files associated with a Unicenter CA-APAS logical output file may be forced from a Unicenter CA-APAS Insight terminal by executing a FLIPXY command with the appropriate ddname/filename in the REQ field and the Adabas session identified in the DBID field. It does not matter whether the entered ddname ends with the letter 'X' or the letter 'Y' -- the action is the same in either way.

Use of the FLIPXY command should be limited to persons who are responsible for the files affected.

Further information about using dual output files is provided in the *Unicenter CA-APAS Systems Guide*.

Turning CA-SpaceMan I/O Logging On or Off

When the CA-SpaceMan DBGIOR5 module is installed with an MPM, Unicenter CA-APAS IOLOGSW command can be used to turn DBGIOR5 I/O logging ON or OFF. This allows I/O logging activity to be limited to the particular time periods desired, thus avoiding the overhead of the I/O logging process at all other times.

To use this command, enter IOLOGSW in the CMD field and either ON or OFF in the REQ field, and press the ENTER key.

Unicenter CA-APAS Zap Status

To get an online display that shows which Unicenter CA-APAS zaps are applied and what their level numbers are, do the following, in the order shown:

1. START request ZAPDECL.

This declares a set of fields that are referenced in the VERZAPS request. Once declared, these field definitions remain in effect for the duration of the MPM session.

If an attempt to START ZAPDECL fails, it is most likely that the fields have already been declared earlier in the MPM session - try proceeding to the next step.

2. START request VERZAPS.

If VERZAPS fails to syntax check, it is probably because the fields that are declared in request ZAPDECL are not in effect. See Step 1, above.

3. Type APASZAPS in the CMD field and press ENTER. Only those zaps that are applied are shown on the screen.

Adabas Functions

Refer to Unicenter CA-APAS Insight online HELP selections for information about the Adabas Functions Menu options.

Resolving Errors

During a Unicenter CA-APAS Insight session, errors may be reported by Natural or by Unicenter CA-APAS.

Errors Reported by Natural

Handling errors from Natural facilities should, in general, be no different for Unicenter CA-APAS Insight than for any other application running under Natural. Natural documentation and your usual in-house or Software AG support staff, as appropriate, should be consulted. If problems in the functioning of Unicenter CA-APAS programs seem indicated, contact Computer Associates technical support.

Syntax Errors in Unicenter CA-APAS Requests

Syntax errors returned by ADD and CHECK commands should be resolved using *Writing Requests* guide which gives the syntax rules for requests. Some common problems and suggested solutions are:

Problem	Solution
Report Line Width Exceeded	Reduce the number and/or width of the columns of output data specified in the request, or increase the report line width by means of the LS parameter within the request.
No INSIGHT-LINES Specified	INSIGHT-LINES must be specified in each request from which data is to be returned to Unicenter CA-APAS Insight terminals.
Invalid Field in SUMMARIZE Request	Fields that do not represent numeric quantities may not logically be summarized. Examples are JOBNAM, TERMINAL-ID, and COMMAND-ID. Such fields may not be included in the summary function list of a SUMMARIZE request. They may, however, be used as WHERE and BY fields, in which case they are used for identification rather than quantification.

Invalid Commands or Operands

Invalid values for individual screen entry fields (CMD, REQ, DBID, SMPID) or invalid combinations of values are reported in the top line of the screen. Correct the value(s) in question and retry the operation. A list of valid command values is available through the Unicenter CA-APAS Insight HELP Facility.

Scrolling Errors

Invalid scroll amounts require correction before scrolling can be done. Valid amounts are: FULL, F, HALF, H, or a numeric value representing the number of columns or lines. When no further scrolling in a given direction is possible, the information message "SCROLL LIMIT REACHED" is displayed.

Terminal Screen Size Errors

You are advised at the beginning of a Unicenter CA-APAS Insight session if the Natural parameters reflecting the number of lines (PS) or the line length (LS) for your terminal indicates an incompatibility with the Unicenter CA-APAS Insight screen formats. If feasible, the Unicenter CA-APAS Insight session continues, otherwise it terminates.

Data Collector Abnormal Terminations

Abnormal termination (abend) of a Data Collector module does not stop the Unicenter CA-APAS Insight terminal interface programs at your terminal. The abended Data Collector module is automatically restarted in most cases, so you should be able to continue your Unicenter CA-APAS Insight session.

If a Data Collector or syntax checker abnormal termination is triggered by a Unicenter CA-APAS request you submitted in a CHECK or ADD command, then you should either identify and change the critical item in the request or discontinue submitting the request.

Other Errors

Various other errors that may occur are reported by a message in the top line of the screen. Invoke the Unicenter CA-APAS Insight HELP Facility to obtain more detailed information if you don't understand the message.

Effective Use of Unicenter CA-APAS Requests

A few techniques of request coding and use that should be helpful with Unicenter CA-APAS are suggested in this section. Other Unicenter CA-APAS guides contain additional information and suggestions. Of course, your own experience is the most fruitful source of effective techniques.

Report Dimensions

The limited viewing area of terminal screens and urgency to quickly focus on key information when working online make it worthwhile to give some thought to the width, quantity and sequencing of report lines generated for online viewing.

Report Line Width

Where feasible, make report lines short enough to fit within your screen width in order to eliminate any need for horizontal scrolling. Some ways to limit the width of output lines include:

- Eliminate any output fields whose usefulness is questionable
- Stack two or three fields within a single column where interpretation of the report is not hindered
- Use override format/length specifications to reduce the display width allocated to fields in cases where you are sure the data values generated fit within the override length

Where report lines exceed the screen width, control the position of fields within the lines to minimize the need for horizontal scrolling:

- Plan to do horizontal scrolling in units of full or half screen widths, using the FULL (F) or HALF (H) scroll amount values rather than specified numbers of columns.
- Try to position fields that must be viewed simultaneously for proper interpretation so that the entire set is displayed in the first display screen or in one of the following screens after FULL or HALF horizontal scrolling.
- Try to position fields that require viewing only in exception cases in the trailing part of a line, beyond the part displayed in the first screen.

- Try to position one or more identifying or reference fields to appear on consecutive screens when horizontal scrolling is used. For instance, depending on the overall report line width, it is sometimes possible to have a pair of fields such as file number and command code show as the rightmost two fields of the first display screen and then show as the leftmost two fields after scrolling FULL to the right. Another approach is to output such fields at two or more places within the line.

Quantity of Report Lines

Several important factors are affected by the number of lines (INSIGHT-LINES) a request causes to be generated:

- Amount of space in the Adabas region that is required to hold data being accumulated for summary requests
- Amount of space in the Adabas region that is required to hold output lines for a request prior to their being transferred to the Unicenter CA-APAS Insight terminal interface
- Amount of space required to hold a set of the transferred lines within the Unicenter CA-APAS Insight terminal interface
- Amount of time and scrolling activity required to view all of a returned set of lines or to find lines of interest within a set of lines

A general guideline is to try to define requests so that you get the information you seek with a minimum number of lines being generated, transferred, buffered, and displayed. Some considerations and techniques are:

- Use the WHERE clause to exclude unneeded Command Log records from a report.
- Use as few BY fields as feasible. The more BY fields, the more unique entity ID combinations to increase space requirements and the number of lines generated.
- Suppress subtotalling (SUBTOT=OFF) in summary requests where feasible.
- Omit the TITLE clause to eliminate a title line.
- Use override column headers to limit column headings to a single line.
- Use SORT DESC where appropriate to cause the lines of most interest to be at the top of the report. This may eliminate the need to vertically scroll past the first screen displayed for the request. It may make it possible to use a very low value for INSIGHT-LINES.
- Keep the value of INSIGHT-LINES as low as possible; where feasible, limit INSIGHT-LINES to the number of lines that fit within one viewing window, either single or dual window size, to eliminate the need for vertical scrolling.

The number of lines that can be transferred is limited by the value of the ADARUN parameter, LU. INSIGHT-LINES should usually not exceed 200.

Request Objectives

Attention to the kinds of requests used and display sequences can speed analysis of performance. The following suggestions may be useful; however, evaluate your requirements and modify requests accordingly.

Determining Causes of Current Poor Performance

One useful approach is to use different types of reports to display information at different levels of detail. For instance, when trying to find out the cause of reported degradation in online response, it might be effective to follow a sequence such as the following:

- Execute one or more SNAP commands to get an overall view of current Adabas nucleus activity as reflected by command and I/O counts, command and I/O rates, average and maximum durations and enqueue times of commands, format translation activity, I/O buffering efficiency, and Hold Queue usage.
- If the SNAPSHOT display indicates that the Adabas Hold Queue is full, the user shown as having the maximum number of holds in effect is a good candidate to eliminate via an Adabas STOPU operator command. In this case, the response problem is likely to be resolved as soon as the offending user has been purged by Adabas.
- If total command or I/O counts in the SNAPSHOT display seem high, one or more summary requests could be used to relate major segments of the workload to particular Adabas file numbers, user types, batch jobs, etc. The SORT DESC feature eliminates the need to look at more than the first one to five lines of these summary reports. Some predefined requests that might be suitable are FILLOAD (by file), UTYPLOAD (by user type), JOBLLOAD (by job name), NATLOAD (by Natural module-id), or TRMLOAD (by terminal-id).
- If total command or I/O counts in the SNAPSHOT display did not seem high enough to account for response degradation, it may be helpful to use some requests that tend to identify likely bottleneck conditions, namely, high update activity or complex searches being enqueued. CTYPLOAD and ENQUEUE are examples of predefined requests suitable for this purpose.

- If either of the above two steps indicates a general source of the performance problem, you have identified entities or types of processing to analyze further at a more detailed level with more selective summary or detail requests. Some predefined requests that could be used or serve as models for this purpose are CLXTRACE (list of each complex search command), FILTRACE (list of each command for a selected file), JOBTRACE (list of each command for a selected job), and the other TRACE requests.

Exception Monitoring

The predefined request RSPCODES is an example of a type of request that could be allowed to run indefinitely. It lists each command that receives a selected response code. Whenever desired, its output could be displayed to see the latest set of commands that had received the specified response codes. The processing overhead of such a request, as well as the amount of output generated for viewing, should normally be quite low.

A request to detect and list commands meeting any other exception condition could be allowed to run indefinitely to enable monitoring the condition from a terminal whenever desired.

Of course, requests for exception reporting may be suspended and restarted, or may be deleted and later added again if the requests do not need to be continually active. For convenience, any requests that are to be run throughout every Adabas session should be started from the Unicenter CA-APAS input stream rather than from the Unicenter CA-APAS Insight terminal interface.

Useful Request Options and Fields

As a reminder, some specific options in the request language that offer considerable power to focus requests include:

WHERE

To include or exclude commands based on specified conditions

BY

To subdivide workload by simple or complex categories

SORT DESC

To position heavy usage items at the top of reports

Some fields that are often most helpful include:

- COUNT of commands and PERCENT of commands
- TOTAL-IO (SUM, PCT, MEAN, MAX)
- DURATION (MEAN, MAX)
- ENQ-TIME (MEAN, MAX)

Batch Execution of Unicenter CA-APAS

When Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS) is executed from within a batch Natural job, a batch oriented Unicenter CA-APAS module is automatically invoked instead of the online modules. The batch module supports a subset of the online commands as shown below.

Supported Commands

Unicenter CA-APAS Commands	Natural Commands
ADD	
CHECK	EDIT
DELBYNAM	LIST
DELBOWN	READ
DELETE	SAVE
END	
FLIPXY	
IOLOGSW	
PAUSE	
Q	
QUIT	
RESET	
RESUME	
SHOW	
START	
.	

END, Q, QUIT and the period, ., are for terminating Unicenter CA-APAS.

The above commands, and any related operands, are provided by the user in card-image record format, one command per record, in the batch Natural input stream. Commands and operands must be in the following fixed format if Natural INPUT statements are being executed in FORMS mode:

col. 1- 8	-	command, left justified
col. 9-16	-	request name or parameter, left justified
col. 17-21	-	database-id number, if other than default
col. 22-23	-	SMPID number, if other than the default zero

Performance History System

The Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS) Performance History System is a comprehensive facility for capturing, storing, consolidating, and reporting Adabas performance data over extended time periods. Use of this history system is optional; it isn't required for use of any of the other facilities previously described in this document. In place of (or in addition to) this history system, Unicenter CA-APAS may be used to generate data for other history systems.

The system as supplied should satisfy the basic needs for performance and capacity planning of most organizations. However, it has been designed as an open system that may easily be modified or extended by experienced DBA personnel. The History file is a standard Adabas file, and all programs that reference the file are written in Natural with source code provided. Detailed information about this system is provided in the *Performance History System* guide.

Areas of Application

Uses for the Performance History System include the following:

- Retrospective Analysis: determining what users were active during a past time period and which ones imposed the greatest loads
- Capacity Planning: determining rates of growth in system use and predicting when capacity will be exhausted
- Load Balancing: finding Adabas applications that are competing for resources in the same time-frame and better scheduling the use of system resources
- Tuning: determining the long-term effects of efforts at tuning the Adabas nucleus and application systems

- System Performance: correlating Adabas performance data with system-generated performance statistics to determine the system resource requirements of a given level of Adabas activity
- Application Performance Prediction: using command duration characteristics of existing applications to project performance of planned applications that will use similar file designs and processing logic

Components

The major components of the Performance History System are briefly described below.

Inputs

The primary input to the History file is data summarized by Unicenter CA-APAS across multiple Adabas commands. This may be supplemented by data from non-Adabas sources. Each of these is discussed below.

Command Data

A standard set of SUMMARIZE requests produce the most commonly required data items for user-specified time intervals, typically fifteen minutes, thirty minutes or one hour. Data may be summarized by any or all of the following:

- Adabas command code
- Adabas file number
- Batch jobname
- Natural library and program name
- Terminal-Id
- CICS transaction code

The chosen requests are submitted as part of the regular input stream for executions of the Data Collector. The requests write the interval statistics to sequential output files. Dual output files provide the ability to capture history data from extended MPM sessions.

User-Supplied Data

The History file includes a number of fields designed to accept supplemental data not available to Unicenter CA-APAS. One example is data produced by job step accounting facilities. Additional fields may be defined by an organization if desired. This category of information may be posted to the History file by online data entry or via batch programs updating the History file. Examples of both types of update programs are included in the Natural source code provided with the system.

History File

A single Adabas file serves as a repository for all historical Adabas performance data generated by the summary requests or provided from other sources by users. Different record types are used to maintain data of different types and at different levels of summarization.

The lowest level of information is for user-specified time intervals within individual Adabas sessions. Sessions are uniquely identified within the History file. Optionally, interval data from the sessions is automatically consolidated to hour, day, week, and/or month levels to provide higher-level pictures of Adabas workloads.

Any combination of Adabas nucleus or utility sessions run on one or more CPU's against any number of databases can be recorded on the single file. MPM, single-user, utility, update, read-only, parallel and stand-alone sessions are accommodated.

To conserve file space, only basic counts, sums and maximums are maintained in the History file. Averages and additional types of summarization are developed by inquiry and reporting programs. Additional record type values and field names have been reserved for local additions to the History file definition.

History Programs

A set of Natural programs performs the following functions:

- Adding new interval summary information to the History file, and, optionally, consolidating interval data to hour, day, week, and month levels
- Performing a number of maintenance functions that help the DBA assess the status of data in the History file and set parameters that governs addition and retention of different types of data in the file
- Purging outdated information according to user-set criteria
- Providing a variety of online displays of performance history data for user-selected time periods, types of information, and entities

Sample Inquiries

A typical sequence of inquiry panels is provided below to give an idea of one of the many uses of the History System. The objective in this example is to find out when the heaviest processing occurred and what caused it. The sample panels illustrate only a few of the options available.

Hourly Workload Panel

The following panel is the result of making menu choices and specifying selection criteria that ask for hourly workloads, in terms of Adabas command counts, across a specified time period.

```

NUMBER OF ADABAS COMMANDS PER HOUR (10,000'S)
ALL COMMANDS AVERAGED FROM 2002-01-01:00 THRU 2002-07-31:23
HH 1 2 3 4 5 6 7 8 9 10 11 12 13 14
+. . . +. . . +. . . +. . . +. . . +. . . +. . . +. . . +. . . +. . . +. . . +. . . +. . .
|
00 | *****
01 | *****
02 | *****
03 | *****
04 |
05 |
06 | *****
07 | **
08 | *****
09 | *****
10 | *****
11 | *****
12 | *****
13 | *****
PRESS PF12 TO VIEW HEAVIEST USERS FOR ANY ONE HOUR

```

Finding hour 9 to be a time of major Adabas processing, pressing PF12 leads to prompting for identifying heaviest entities responsible for the processing loads during hour 9. The following panels show the heaviest file numbers, batch jobs, Natural User-Ids (application/library-id), and Natural programs during hour 9.

Five Heaviest Adabas File Numbers Panel

The following panel lists the five heaviest used files by command count.

5 HEAVIEST ADABAS FILE NUMBERS BY CMD CNT FOR HOUR 09 TOTAL USE FOR PERIOD 2002-01-01:00 THRU 2002-07-31:23								
RNK	FNR	CMD CNT	TOT IO	TOT CPU	MEAN IO	MEAN CPU	MEAN DUR	MEAN ENQ
1	11	499,814	44319	103277.78	0.09	0.20663	0.00000	0.00000
2	96	28,375	22152	9758.21	0.78	0.34396	0.00000	0.00000
3	30	7,437	1871	1960.05	0.25	0.25549	0.00000	0.00000
4	26	7,371	9861	2677.17	1.34	0.36320	0.00000	0.00000
5	98	7,265	9954	2318.93	1.37	0.31919	0.00000	0.00000
AVG		20,691	5944	4982.74	0.29	0.24082	0.00000	0.00000

Five Heaviest Adabas Batch Jobs Panel

The following panel lists the five heaviest Adabas batch jobs by command count.

FIVE HEAVIEST ADABAS BATCH JOBS BY CMD CNT FOR HOUR 09 TOTAL USE FOR PERIOD 2002-01-01:00 THRU 2002-07-31:23					
JOB NAME	CMD CNT	TOT IO	TOT CPU	MEAN IO	MEAN CPU
A30N0107	37888	84679	65965.88600	2.23	1.74108
D10T0502	31191	22127	8342.13400	0.71	0.26745
P25ZR405	25314	3706	7930.12100	0.15	0.31327
\$M104100	20691	6201	7769.58700	0.30	0.37551
\$D1119JS	2423	2670	1202.33200	1.10	0.49622
AVERAGE	9851	10050	7643.46367	1.02	0.77591

Five Heaviest Natural User IDs Panel

The following panel lists the five heaviest Natural user-ids by command count.

FIVE HEAVIEST NATURAL USER-IDS BY CMD CNT FOR HOUR 09 TOTAL USE FOR PERIOD 2002-01-01:00 THRU 2002-07-31:23					
NATURAL USER-ID	CMD CNT	TOT IO	TOT CPU	MEAN IO	MEAN CPU
TRSJ	32229	22988	8510.09600	0.71	0.26405
SYSLIB	4449	5641	960.53800	1.27	0.20241
DORBMF	630	4508	1694.25800	7.16	2.68930
DORWIT	215	2116	299.07200	9.84	1.39103
DELJIS	58	11	3.58400	0.19	0.06179
AVERAGE	3438	3258	1043.93600	0.95	0.30365

Five Heaviest Natural User Programs Panel

The following panel lists the five heaviest Natural user programs by command count.

FIVE HEAVIEST NATURAL USER PROGRAMS BY CMD CNT FOR HOUR 09 TOTAL USE FOR PERIOD 2002-01-01:00 THRU 2002-07-31:23						
NATURAL USER-ID	PROGRAM	CMD CNT	TOT IO	TOT CPU	MEAN IO	MEAN CPU
-----		-----	-----	-----	-----	-----
TRSJ	TXNTR03	10952	8648	2962.23200	0.79	0.26500
TRSJ	TXINQ01	5954	7177	1961.54000	1.21	0.32945
TRSJ	TXUPD01	4924	2945	1552.84800	0.60	0.31536
TRSJ	TSRPT22	4341	778	972.64000	0.18	0.22406
SYSLIB	SYSFULL	3828	4808	806.70200	1.26	0.21074
AVERAGE		600	569	182.27454	0.95	0.30379

Sample Unicenter CA-APAS Insight Session

This appendix describes the user actions and shows the resulting panel displays for a sample Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS) session. Many of the basic functions available in Unicenter CA-APAS are demonstrated. For explanations of additional functions, see the chapter “Using Unicenter CA-APAS Online.”

Invoking Unicenter CA-APAS Insight from Natural

Invoke Natural and log on to the library-id or application that contains the Unicenter CA-APAS Insight source and object modules. The usual library-id or application is INSIGHT, but it could be different at your site, as it is for this sample session.

NEXT INSIGHT

LIB=INS410

At the NEXT prompt, invoke the Insight system by typing **INSIGHT**, as shown above, and pressing ENTER.

Depending how your Natural environment is set up when you invoke Natural, you could receive just the NEXT prompt. Alternatively, the Natural Application Development Main Menu, or another site-customized panel, might be displayed.

Initial Insight Panel

The panel below indicates that Insight is ready to accept commands or provide HELP information.

```

      ADABAS PERFORMANCE ANALYSIS SYSTEM (CA-APAS)
    COPYRIGHT 2003 COMPUTER ASSOCIATES INTERNATIONAL, INC.
  ALL RIGHTS RESERVED.  CONTAINS TRADE SECRETS AND CONFIDENTIAL PROPRIETARY
    INFORMATION OF COMPUTER ASSOCIATES INTERNATIONAL, INC.
  REVERSE ENGINEERING PROHIBITED.  COPYRIGHT NOTICE DOES NOT IMPLY PUBLICATION.

  IIII      NNN      NN      SSSSSSS      IIII      GGGGGGGG      HH      HH      TTTTTTTT
  II        NNNN     NN      SS              II        GG              HH      HH      TT
  II        NN NN     NN      SS              II        GG              HH      HH      TT
  II        NN NN     NN      SSSSSSS      II        GG              HHHHHHHH      TT
  II        NN      NN NN      SS              II        GG      GGGG      HH      HH      TT
  II        NN      NNNN      SS              II        GG      GG      HH      HH      TT
  IIII      NN      NNN      SSSSSSS      IIII      GGGGGGGG      HH      HH      TT

-----
I N T E R A C T I V E      A D A B A S      P E R F O R M A N C E      M O N I T O R
-----

      CA-INSIGHT V4.01.01 FOR ADABAS V7 & NATURAL V3

CMD: _____ REQ: _____ DBID: ____0 SMPID: _0 > USE PF KEY, CMD, OR ENTER <
PFK: 1=HELP 3=MENU 4=SELECT 5=SHOW 6=DISPLAY 9=SNAP 12=REQMGT INSTART
  
```

Press ENTER to display the Main Menu screen.

Main Menu Panel

The Insight Main Menu panel is primarily for the benefit of those unfamiliar with Insight and those who use it infrequently. More experienced users usually skip this panel.

```

C A - I N S I G H T      M A I N      M E N U

CODE  FUNCTION                                                    (COMMAND)
-----
1    INFORMATION ABOUT CURRENT CA-APAS REQUESTS  (SHOW)
2    DISPLAY DATA FROM CURRENT REQUEST(S)      (DISPLAY)
3    SELECT A SAVED REQUEST TO START OR EDIT    (SELECT)
4    REQUEST MANAGEMENT MENU

5    TRACE      - (GENERATE AND START NEW REQUEST)
6    SUMMARY    -      ''
7    EXCEPTION  -      ''

8    ADABAS NUCLEUS PROCESSING STATISTICS      (SNAP)
9    ADABAS FUNCTIONS MENU                      (ADAMENU)
H    HELP ABOUT CA-INSIGHT
.    TERMINATE CA-INSIGHT
A    CA-APAS HISTORY APPLICATION
S    CA-SPACEMAN APPLICATION

CODE:  _      USE MENU CODE, PF-KEY, OR CA-INSIGHT COMMAND

CMD:  _____ REQ:  _____ DBID:  _____ 0 SMPID:  _____ 0 DT: 02-11-21 TM: 22:36:51.0
PFK:  1=HELP  3=END  4=SELECT  5=SHOW  6=DISPLAY  9=SNAP  12=REQMGT  INMENU

```

Many of the basic functions of Insight are available from this screen via simple menu selection codes. Automatic generation and activation of new traces, summary or exception requests, along with selection of any request already saved in the Insight library, enables users to take advantage of the power and flexibility of the request facility without any need to learn request syntax.

To browse a list of previously saved requests in the library, type the number **3** in the CODE field and press ENTER.

Alternatively, type **SE** in the CMD field of any Insight panel and press ENTER.

SELECT Panel

The SELECT function lets users browse through a listing of the names and descriptions of Unicenter CA-APAS requests that reside in the Natural system file. Predefined requests that are distributed with Unicenter CA-APAS as well as any saved by users are listed as shown below.

```

      R E Q U E S T S      I N      C U R R E N T      L I B R A R Y

ACBXMISS * COMMAND LOGS WITHOUT ACBX INFORMATION
ACBXVRFY * USED TO VERIFY ACBX INSTALLATION UNDER ANY ADABAS VERSION
ADARUN   * SPECIAL REQUEST WHICH MUST BE ACTIVE FOR "ADARUN" COMMAND
ALLCMDS  * OUTPUT FILE SUMMARY OF ALL COMMANDS FOR QUARTER HOUR INTERVALS
ALLTRACE * LIST ALL INDIVIDUAL COMMANDS
BFLADA   * SUMMARY OF I/O BUFFER FLUSH COUNTS FROM ADABAS
BFLUSHES * STATISTICS FOR ASSUMED ADABAS I/O BUFFER FLUSHES, BY CMD CODE
BFLX     * "DECLARE" A BUFFER-FLUSH INDICATOR, BFLX
BIGFIND  * IDENTIFY SEARCH COMMANDS QUALIFYING EXCESSIVE NUMBERS OF RECORDS
BYMINUTE * GRAPHIC DISPLAY OF COMMAND COUNTS BY MINUTE OF DAY
CASSOXXX * SPACEMAN ONLY - MOST READ ASSO RABNS FOR ADARUN CASSO... PARMS
CDATAXXX * SPACEMAN ONLY - MOST READ DATA RABNS FOR ADARUN CDATA... PARMS
CLOSTIME * REPORT TOTAL RESOURCE USAGE FROM THE 'CL' COMMAND
CLXTRACE * LIST ALL INDIVIDUAL "COMPLEX" SEARCH COMMANDS
CMDSUM   * PERCENT OF CMDS & I/O BY COMMAND CODE WITHIN FILE NUMBER
CTYPLOAD * SUMMARY OF LOADS BY COMMAND-CODE WITHIN COMMAND-TYPE
CWORKFAC * SPACEMAN ONLY - MOST READ WORK RABNS FOR ADARUN CWORK.FAC PARMS
DATALOC  * MAY REFLECT RELATIVE DATA I/O SERVICE TIMES BETWEEN FILES
ARK 1 REQUEST: E-EDIT S-START -OR- TO RESTART BROWSE, PUT START VALUE IN REQ:
MD: _____ REQ: _____ DBID: _____ SMPID: _____ DT: 02-11-21 TM: 22:38:59.6
FK: 1=HELP 3=MENU 5=SHOW 6=DISPLAY 9=SNAP 12=REQMGT INSELECT

```

To reposition the browse of request names at the first request whose name begins with the letter 'f', type the letter f into the REQ field and press ENTER.

SELECT Panel - After Browse

The SELECT panel is automatically scrolled to the first request name beginning with the letter 'f'.

```

      R E Q U E S T S      I N      C U R R E N T      L I B R A R Y

FBEFFICY * BASELINE STATISTICS FOR USE IN TUNING ADABAS LFP PARM
FBRATIO  * POTENTIAL VS. ACTUAL FORMAT TRANSLATIONS
FBTRACE  * SHOWS FORMAT BUFFERS FOR SELECTED COMMANDS
FILCMDIO * PERCENT OF CMDS & I/O BY COMMAND CODE WITHIN FILE NUMBER
FILCMDS  * OUTPUT FILE SUMMARY OF COMMAND PROCESSING BY FILE NUMBER
FILFBUSE * SHOWS WHICH PROGRAMS USE A GIVEN FIELD IN FORMAT BUFFERS
FILLOAD  * SUMMARY OF FILE LOADS IN DESCENDING ORDER
FILTRACE * LIST INDIVIDUAL COMMANDS FOR A SELECTED FILE
GENSUMXY * SAMPLE "PRINT TO DUAL" OUTPUT FILE - MODIFY BEFORE USE
HDTRAVEL * SPACEMAN ONLY - DASD VOLUMES IN DESC ORDER OF AVG HEAD TRAVEL.
HIWATER  * SPECIAL REQUEST - MUST BE ACTIVE FOR HIWATER CMD
HOLDQ    * FIND OUT WHICH USER HAS MOST RECORDS HELD IN AN INTERVAL
HOUR-MIN * "DECLARE" TO ALLOW REPORTING BY HOUR/MINUTE OF DAY
IFBLEXTR * COMPARE FBL IN ACBX WITH ALLOCATED IFBL SIZE
IFBLSUM  * DETERMINE THE MIN/MEAN/MAX FBL, IFBL, IFBL-ALLOC
IOBYVOL  * SPACEMAN ONLY - REPORT READS FOR CACHE DETERMINATION
IOLOAD   * SPACEMAN ONLY - I/O COUNTS BY RABN RANGE W/IN DATASET
IOSET    * SPACEMAN - IDENTIFY SIZES OF WORKING SETS OF ASSO & DATA BLOCKS
MARK 1 REQUEST: E-EDIT S-START -OR- TO RESTART BROWSE, PUT START VALUE IN REQ:
CMD: _____ REQ: _____ DBID: _____ SMPID: _____ DT: 02-11-21 TM: 22:39:46.4
PFK: 1=HELP 3=MENU 5=SHOW 6=DISPLAY 9=SNAP 12=REQMGT INSELECT
  
```

To start FILLOAD, type the letter s (for START) in the selection field for request FILLOAD and press ENTER.

After the ADD of FILLOAD has been done, press PF5 or type the letters **sh** in the CMD field to invoke the SHOW display.

SHOW Panels

The SHOW screen lists the requests that are currently known to the Data Collector and selected status items about those requests. The following panel is an example of the display.

STATUS OF APAS REQUESTS FOR DBID: 0										LINE	1 OF	25
REQ NAME	TY	INTVL	ST	INLN	TSIZE	PRINT TO	OUT-FILE	OWNER	SEC			
d FILLOAD	S		A	0100	00003			*	N			
USERINFO	D		A	0100				*	N			
UPDSAT	S		A		00003	GENSUM		*	N			
MAXIUBL	S		A	0005	00001	GENSUM		*	N			
SRCHDES	S		A		00003	GENSUM		*	N			
ADAMEFF	S		P		00012	GENSUM		*	N			
RECLEND	S		A		00003	GENSUM		*	N			
RECLENC	S		A		00003	GENSUM		*	N			
L3SEQ	S		A		00005	GENSUM		*	N			
BFLUSHES	S		A		00004	GENSUM		*	N			
TIOSUM	S		A		00007	GENSUM		*	N			
DESCUPD	S		A		00014	GENSUM		*	N			
THRDDUR	S		A		00009	GENSUM		*	N			
THRDSUM	S		A		00008	GENSUM		*	N			
NATUSE	S		A		00008	GENSUM		*	N			
NATSUM	S		A		00023	GENSUM		*	N			
TRMSUM	S		A		00003	GENSUM		*	N			
MARK 1 REQUEST: D-DISPLAY -OR- ANY REQ(S): R-RESET P-PAUSE U-RESUME X-DELETE												
CMD: REQ: DBID: 0 SMPID: 0 DT: 02-11-21 TM: 22:40:44.6												
PFK: 1=HELP 3=MENU 4=SELECT 6=DISPLAY 9=SNAP 12=REQMGT INSHOW												

The status information about the requests is as follows:

Heading	Description
TY	Indicates the type of request; S for summary, D for detail, or C for COPY.
INTVL	Indicates the interval length in seconds where applicable.
ST	Indicates the operational status; A for active or P for paused.
INLN	Indicates the number of lines available for display at Unicenter CA-APAS Insight terminals
TSIZE	Indicates the amount of virtual storage for summary tables in kilobytes.
PRINT TO	Lists the names of files to which print formatted report are sent.
OUT-FILE	Lists the names of files to which non-print formatted data are sent.

Heading	Description
OWNER	Specifies the ID of person or group that owns the request. An asterisk, *, indicates a public request.
SEC	Indicates the security level of the request; N for none, M for modify, or D for display.

One or more of the listed requests may be displayed, deleted, etc., simply by typing the appropriate codes beside the request name.

To display a FILLOAD request, type the letter **d** next to FILLOAD and press ENTER.

FILLOAD Panel

The data lines from FILLOAD are displayed, as shown below.

```

DISPLAY FILLOAD WAS DONE
DBID 00003 REQ FILLOAD DT 2002-11-21 TIME 17:40:40 TO 17:41:38 LINE 1 OF 5
FIL  COMMANDS  % OF TOTAL      TOTAL IO  % OF TOTAL      RATE
-----
  0      584 |=====      28 | =      0.56
150      399 |=====      217 | =====      4.34
 11      170 |====      24 | =      0.48
  4       44 | =      0 |      0.00
  8        2 |      2 |      0.04

```

```

CMD: _____ REQ: FILLOAD DBID: _____ SMPID: 0 DT: 02-11-21 TM: 22:41:39.5
PFK: 3=MENU 4=SELECT 5=SHOW SCROLL: _____ 7=U 8=D 10=L 11=R INDIS080

```

To access a different Insight function, press PF3 to return to the Main Menu panel .

Generate Trace Request Panel - 1 of 2

To automatically generate and activate a Unicenter CA-APAS Trace request, type the number 5 in the Main Menu CODE field. After you press ENTER, the following prompt panel appears:

G E N E R A T E T R A C E R E Q U E S T			
---	--	--	--

1. MARK THE FIELDS TO BE REPORTED:

<input type="checkbox"/> CMDSEQ <input checked="" type="checkbox"/> DATEJ <input checked="" type="checkbox"/> DATE <input checked="" type="checkbox"/> TIME <input checked="" type="checkbox"/> JOBNAME <input type="checkbox"/> JOBNAME2 <input type="checkbox"/> JOB-STEP <input type="checkbox"/> JMR-USER-ID <input type="checkbox"/> USER-ID <input type="checkbox"/> GLOBAL-VM-ID <input type="checkbox"/> GLOBAL-USER-ID <input type="checkbox"/> TERMINAL-ID <input type="checkbox"/> TID <input type="checkbox"/> USER-TYPE <input type="checkbox"/> USER-PRIORITY <input type="checkbox"/> IUBL	<input type="checkbox"/> TP-TRANS-NAME <input type="checkbox"/> TP-TRANS-ID <input checked="" type="checkbox"/> PROGRAM <input type="checkbox"/> NAT-LOGON <input type="checkbox"/> NAT-PROG <input type="checkbox"/> NAT-STMT-NO <input checked="" type="checkbox"/> THREAD <input type="checkbox"/> CMD <input type="checkbox"/> CMD-TYPE <input type="checkbox"/> FILE <input type="checkbox"/> RSP <input type="checkbox"/> DE-UPD <input type="checkbox"/> ASSO-IO <input type="checkbox"/> DATA-IO <input type="checkbox"/> WORK-IO <input checked="" type="checkbox"/> TOT-IO	<input type="checkbox"/> CPU <input type="checkbox"/> ENQ-TIME <input type="checkbox"/> DURATION <input type="checkbox"/> ENQ-DUR <input type="checkbox"/> ECBS <input type="checkbox"/> HOLD-COUNT <input type="checkbox"/> ACB12 <input type="checkbox"/> CID <input type="checkbox"/> COPT1 <input type="checkbox"/> COPT2 <input type="checkbox"/> ADDITIONS-1 <input type="checkbox"/> ADDITIONS-2 <input type="checkbox"/> ADDITIONS-5 <input type="checkbox"/> USER-AREA <input type="checkbox"/> FB-ERR-FIELD <input type="checkbox"/> IFBL	<input type="checkbox"/> FBL <input type="checkbox"/> RBL <input type="checkbox"/> SBL <input type="checkbox"/> VBL <input type="checkbox"/> IBL <input type="checkbox"/> ISN <input type="checkbox"/> ISNLL <input type="checkbox"/> ISNQ <input type="checkbox"/> REC-LENGTH-COMP <input type="checkbox"/> REC-LENGTH-DEC <input type="checkbox"/> USERA <input type="checkbox"/> USERB <input type="checkbox"/> USERC <input type="checkbox"/> USER1 <input type="checkbox"/> USER2 <input type="checkbox"/> USER3
--	---	--	--

CMD: _____ REQ: _____ DBID: _____ 0 SMPID: 0 DT: 02-11-21 TM: 22:43:00.2
 PFK: 1=HELP 3=MENU 4=SELECT 5=SHOW 6=DISPLAY 9=SNAP 12=REQMGT INTRAC

You can mark data fields for inclusion in the report. You can accept or change the default request name shown in the REQ field. Press ENTER.

Generate Trace Request Panel - 2 of 2

Continuing from the prior panel, the second prompt screen appears:

```

      G E N E R A T E      T R A C E      R E Q U E S T

2. SPECIFY COMMAND SELECTION CRITERIA IF DESIRED
   (MULTIPLE CONDITIONS WILL BE AND'ED TOGETHER):

   _ JOBNAME           = _____
   _ JOBNAME2          = _____
   _ GLOBAL-USER-ID    = _____
   x TERMINAL-ID       = 2807
   _ PROGRAM           = _____
   _ NATURAL-LOGON     = _____
   _ NAT-PROGRAM       = _____
   _ TP-TRANS-NAME     = _____
   _ USER-TYPE        = _____
   _ FILE              = _____

   _ RESPONSE-CODE     = ____ TO ____
   _ ASSO-IO           > 25____
   _ DATA-IO          > 10____
   _ WORK-IO           > 10____
   _ TOTAL-IO          > 50____

   _ ENQ-TIME          > 0.500__
   _ DURATION           > 0.500__
   _ EST-CPU-TIME      > 0.015__

3. REPORT TITLE (OPTIONAL): _____

4. CHANGE REQUEST NAME AND/OR DBID (BELOW) IF DESIRED; PRESS <ENTER>

CMD: _____ REQ: A5T4405 DBID: _____ SMPID: 0 DT: 02-11-21 TM: 22:43:00.2
PFK: 1=HELP 2=PRIOR SCREEN 3=MENU 4=SELECT 5=SHOW 12=REQMGT INTRAC

```

Since the intent is to trace all Adabas calls from your own terminal, the TERMINAL-ID field is marked as the only selection criterion.

When you press ENTER, the generated request, which is not shown here, is displayed on your terminal by automatic execution of Natural's LIST command. This provides an opportunity to review the generated request and obtain a hardcopy listing of it by entering the letter **H** in the COMMAND field of the LIST function, or by using any other screen-print facility.

Press ENTER to refresh the panel and display its data up to that point.

To delete the request, type **del** in the CMD field of the request display and press ENTER.

Natural Editor Panel

To edit an existing request, such as NATLOAD, during an Insight session, type **edit** in the CMD field and **natload** in the REQ field of a menu panel. The following Natural editor screen appears:

```

>
All .....1.....2.....3.....4.....5.....6.....7..
0010 * NATURAL PROGRAM LOADS IN DESCENDING ORDER
0020 *
0030 NATLOAD: SUMMARIZE
0040     COUNT (HD='COMMANDS') SORT DESC
0050     PERCENT (HD='% OF TOTAL' PF=G11)
0060     SUM(TOTAL-IO) (HD='TOTAL IO')
0070     PCT(TOTAL-IO) (HD='% OF TOTAL' PF=G11)
0080     BY NAT-MOD-ID (HD='NATURAL MODULE ID')
0090     INSIGHT-LINES = 20
0100     ;
0110
0120
0130
0140
0150
0160
0170
0180
0190
0200
.....1.....2.....3.....4.....5..... S 10  L 1

```

To be able to display more report lines at any one time, you could change the value of INSIGHT-LINES from 20 to 50, as shown next.

```
>                                     > + Program    NATLOAD  Lib INS410
All  .....1.....2.....3.....4.....5.....6.....7...
0010 * NATURAL PROGRAM LOADS IN DESCENDING ORDER
0020 *
0030 NATLOAD: SUMMARIZE
0040      COUNT (HD='COMMANDS') SORT DESC
0050      PERCENT (HD='% OF TOTAL' PF=G11)
0060      SUM(TOTAL-IO) (HD='TOTAL IO')
0070      PCT(TOTAL-IO) (HD='% OF TOTAL' PF=G11)
0080      BY NAT-MOD-ID  (HD='NATURAL MODULE ID')
0090      INSIGHT-LINES = 59
0100      ;
0110
0120
0130
0140
0150
0160
0170
0180
0190
0200 .....1.....2.....3.....4.....5..... S 10  L 1
```

To see the actual NATLOAD data display, type a period at the > prompt to exit the editor, leaving the modified NATLOAD request in the source work area.

Activate the NATLOAD Request

Once you exit the editor, you return to the Insight Request Management panel to start the request. To do this, type **add** in the CMD field to activate NATLOAD. The Request Management panel indicates a successful add in the upper left-hand corner.

```

ADD      NATLOAD WAS DONE ON DBID: 00003
          I N S I G H T   R E Q U E S T   M A N A G E M E N T

USE PFKEY OR TYPE FULL OR ABBR. COMMAND IN CMD; USE REQ & DBID AS SHOWN

ABBR COMMAND  REQ DBID    (O=OPTIONAL, R=REQUIRED)
-----
SH  SHOW      0          - TELLS ABOUT CURRENT REQUESTS WITH OPTION TO ACT ON ANY
SE  SELECT    0          - BROWSE SAVED REQUESTS; MARK ONE TO EDIT OR START
DI  DISPLAY   R  0       - GIVES CURRENT DATA FROM NAMED ACTIVE REQUEST

ST  START     R  0       - READS NAMED REQUEST INTO SOURCE AREA AND ACTIVATES IT
DE  DELETE    R  0       - DELETES NAMED REQUEST AND ITS DATA FROM PROCESSING

ED  EDIT      0          - NATURAL EDIT COMMAND (ANY VALID OPERAND IN REQ)
SA  SAVE      0          - NATURAL SAVE COMMAND (ANY VALID OPERAND IN REQ)
CH  CHECK     0          - SYNTAX CHECKS REQUEST NOW IN SOURCE AREA
AD  ADD       0          - SYNTAX CHECKS & ACTIVATES REQUEST NOW IN SOURCE AREA

LI  LIST      0          - NATURAL LIST COMMAND (ANY VALID OPERAND IN REQ)

CMD: _____ REQ: NATLOAD DBID: _____ SMPID: 0 DT: 02-11-21 TM: 22:48:26.1
PFK: 1=HELP 3=END 4=SELECT 5=SHOW 6=DISPLAY 9=SNAP 12=REQMGT INMENU

```

After a request is added, type **di** in the CMD field to display NATLOAD.

NATLOAD Panel

The date lines from NATLOAD are displayed as shown below.

```

DISPLAY NATLOAD WAS DONE
DBID 00003 REQ NATLOAD DT 2002-11-21 TIME 17:48:25 TO 17:48:25 LINE 1 OF 3
NATURAL MODULE ID      COMMANDS % OF TOTAL      TOTAL IO % OF TOTAL
-----
INS410 INRQMGR M00000      6 |=====      6 |=====
      1 |==              0 |

```

Press ENTER successively to repeat the DISPLAY command and show updated values.

Deleting Requests

If you want to delete the requests you have been using, press PF5 to invoke the SHOW display of active requests, then type the letter x to mark the requests for deletion:

```

STATUS OF APAS REQUESTS FOR DBID:      0
-----
REQ NAME  TY  INTVL  ST  INLN  TSIZE  PRINT TO  OUT-FILE  OWNER  1 OF 26
-----
x NATLOAD  S      A    0050  00003
x FILLOAD  S      A    0100  00003
  USERINFO D      A    0100
  UPDSAT    S      A    00003  GENSUM
  MAXIUBL   S      A    0005  00001  GENSUM
  SRCHDES   S      A    00003  GENSUM
  ADAMEFF   S      P    00012  GENSUM
  RECLEND   S      A    00003  GENSUM
  RECLENC   S      A    00003  GENSUM
  L3SEQ     S      A    00005  GENSUM
  BFLUSHES  S      A    00004  GENSUM
  TIOSUM    S      A    00007  GENSUM
  DESCUPD   S      A    00014  GENSUM
  THRDDUR   S      A    00009  GENSUM
  THRDSUM   S      A    00008  GENSUM
  NATUSE    S      A    00008  GENSUM
  NATSUM    S      A    00023  GENSUM
MARK 1 REQUEST: D-DISPLAY -OR- ANY REQ(S): R-RESET P-PAUSE U-RESUME X-DELETE
CMD:  REQ:      DBID:      0 SMPID:  0 DT: 02-11-21 TM: 22:51:56.6
PFK: 1=HELP 3=MENU 4=SELECT 6=DISPLAY 9=SNAP 12=REQMGT      INSHOW

```

In this sample panel, NATLOAD and FILLOAD have been marked for deletion.

Press ENTER. The requests are deleted and confirmation is displayed:

STATUS OF APAS REQUESTS FOR DBID: 0						LINE	1 OF	26		
	REQ NAME	TY	INTVL	ST	INLN	TSIZE	PRINT TO	OUT-FILE	OWNER	SEC
DELETED X	NATLOAD	S		A	0050	00003			*	N
DELETED X	FILLOAD	S		A	0100	00003			*	N
	USERINFO	D		A	0100				*	N
	UPDSAT	S		A		00003	GENSUM		*	N
	MAXIUBL	S		A	0005	00001	GENSUM		*	N
	SRCHDES	S		A		00003	GENSUM		*	N
	ADAMEFF	S		P		00012	GENSUM		*	N
	RECLEND	S		A		00003	GENSUM		*	N
	RECLENC	S		A		00003	GENSUM		*	N
	L3SEQ	S		A		00005	GENSUM		*	N
	BFLUSHES	S		A		00004	GENSUM		*	N
	TIOSUM	S		A		00007	GENSUM		*	N
	DESCUPD	S		A		00014	GENSUM		*	N
	THRDDUR	S		A		00009	GENSUM		*	N
	THRDSUM	S		A		00008	GENSUM		*	N
	NATUSE	S		A		00008	GENSUM		*	N
	NATSUM	S		A		00023	GENSUM		*	N
<ENTER> TO CONTINUE										
CMD:	REQ:		DBID:	0	SMPID:	0	DT:	02-11-21	TM:	22:51:56.6
PFK:	1=HELP	3=MENU	4=SELECT	6=DISPLAY	9=SNAP	12=REQMGT			INSHOW	

Creating A New Request

If you want to create a new request using the Natural full screen editor, type **edit** in the CMD field of the current Insight panel and press ENTER. The following panel, showing that NATLOAD is still present in the source work area, is displayed:

```

>                                     > + Program   NATLOAD  Lib INS410
All .....1.....2.....3.....4.....5.....6.....7..
0010 * NATURAL PROGRAM LOADS IN DESCENDING ORDER
0020 *
0030 NATLOAD: SUMMARIZE
0040     COUNT (HD='COMMANDS') SORT DESC
0050     PERCENT (HD='% OF TOTAL' PF=G11)
0060     SUM(TOTAL-IO) (HD='TOTAL IO')
0070     PCT(TOTAL-IO) (HD='% OF TOTAL' PF=G11)
0080     BY NAT-MOD-ID  (HD='Natural MODULE ID')
0090     INSIGHT-LINES = 50
0100 ;
0110
0120
0130
0140
0150
0160
0170
0180
0190
0200
.....1.....2.....3.....4.....5..... S 10  L 1

```

Clear the source work area by typing the command **clear** at the > prompt. Press ENTER.

Once the source work area is cleared, type your new request into the source line area.

```

>
All      ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
0010 newreq: summarize
0020          count
0030          percent
0040          total-io
0050          by file
0060          cmd
0070          insight-lines = 100 ;
0080
0090
0100
0110
0120
0130
0140
0150
0160
0170
0180
0190
0200
      ....+....1....+....2....+....3....+....4....+....5....+... S 0    L 1

```

Press ENTER to accept the new lines , which change to all uppercase. Type the **save newreq** command at the > prompt to save this request keyed into the source work area. Exit the editor by typing a period in the EDIT command line.

Syntax Checking the New Request

To syntax check a request without actually starting the request, type **ch** for CHECK in the CMD field and press ENTER.

The following screen is displayed due to a syntax error in the new request. TOTAL-IO is a data field; only summary functions are allowed in this part of a SUMMARIZE request.

```

SYNTAX ERROR - YOU MAY SCROLL UP/DOWN BEFORE NEXT FUNCTION.
DBID      0 REQ      DT      TIME      TO      LINE  1 OF  12
          10 NEWREQ:  SUMMARIZE
          20          COUNT
          30          PERCENT
          40          TOTAL-IO
                        <|
DBG02019E 2002-11-21 17:56:43 INVALID SUMMARY FUNCTION SPECIFICATION
          50          BY FILE
          60          CMD
          70          INSIGHT-LINES=100 ;

DBG04001E 2002-11-21 17:56:43 SYNTAX ERROR IN CA-INSIGHT REQUEST;  NOT ADDED

```

To get back into the editor to correct the request, type **ed** for EDIT in the CMD field and press ENTER.

Correcting a Syntax Error

The editor panel is redisplayed. In this sample, the request was changed to specify the SUM function of the TOTAL-IO data field.

```

>
All .....1.....2.....3.....4.....5.....6.....7..
0010 NEWREQ:  SUMMARIZE
0020          COUNT
0030          PERCENT
0040          sum(TOTAL-IO)
0050          BY FILE
0060          CMD
0070          INSIGHT-LINES = 100 ;
0080
0090
0100
0110
0120
0130
0140
0150
0160
0170
0180
0190
0200
.....1.....2.....3.....4.....5..... S 7    L 1

```

Once you have modified the request, type the **SAVE** command at the > prompt and press ENTER. Exit the editor and add the request.

Displaying a New Request

To display the request you have created, type **di** in the CMD field and **newreq** in the REQ field. This is the resulting display:

```

DISPLAY NEWREQ  WAS DONE
DBID 00003 REQ NEWREQ  DT 2002-11-21 TIME 17:58:30 TO 17:58:35 LINE  1 OF 16
  FIL M      CMD TOTAL      SUM
  NUM D      COUNT CMDS      TOT
  IO
-----
   0 RC      31  30.1      0
   0 ==      31  30.1      0
  10 L3      55  53.4      48
  10 ==      55  53.4      48
  11 L3      16  15.5      16
     L9       1   1.0       2
  11 ==      17  16.5      18
-----
===== ==      103 100.0      66

MD: _____ REQ: NEWREQ DBID: _____ SMPID: _____ DT: 02-11-21 TM: 22:58:37.2
FK: 3=MENU 4=SELECT 5=SHOW SCROLL: _____ 7=U 8=D 10=L 11=R INDIS080

```

Press PF3 to return to the Main Menu.

SNAPSHOT Panel

Start SNAPSHOT by typing the number 8 in the CODE field and pressing ENTER or press the PF9 key from any Insight panel to activate the SNAP display. The SNAPSHOT panel shown below is a special, fixed-format report that gives selected summary and instantaneous data about Adabas nucleus activity and status.

In the first SNAP display following the ADD of SNAPSHOT, the LOG.IO and EFF values are not accurate for the initial SNAP interval.

```

ADD      SNAPSHOT WAS DONE ON DBID: 00003
DBID 00003 REQ SNAPSHOT DT 2002-11-21 FROM 17:59:53 TO 17:59:53
-----
TOT CMD:      2      DUR MAX:      0.000      AVG:      0.0000
  RATE:      2.000      ENQ MAX:      0.0000      AVG:      0.00000
TOT IO:      0      CPU MAX:      0.000      AVG:      0.0000
  RATE:      0.00      IO MAX:      0      AVG:      0.00
INTRPTS:    4,185
-----
  USERS:      3      ECB MAX:      1      AVG:      1.00
-----
BP SIZE:    246,784      %AVAIL:      93.7      %DATA:      68.4
LOG.IO:     8,819      %WRITE:      6.2      %ASSO:      31.5
PHY.IO:      0      %ACTIVE:      0.0
  EFF:     8,819.0
FLUSHES:      0
-----
FB TRAN:      17      FMT POOL SIZE:    200,000      %USED:      0.6
OVRWR:      0      WORK POOL SIZE:    499,744
-----
HQE USE:      0 MAX:      0 BY ID:      X 4040404040404040 JOB:
-----
CMD: _____ REQ: _____ DBID: _____ SMPID: _____ DT: 02-11-21 TM: 22:59:54.2
PFK: 1=HELP      3=MENU 4=SELECT 5=SHOW 6=DISPLAY 12=REQMGT INSNAP

```

To repeat the SNAP for a subsequent interval, press ENTER. The LOG.IO and EFF values are accurate for this and all subsequent intervals.

Detailed descriptions of the data items and their interpretation are available through the Insight online HELP facility. Briefly, the SNAPSHOT display is useful for making the following kinds of assessments about current activity in a given copy of the Adabas nucleus:

- The magnitude of the overall workload being imposed by application
- Whether applications are being held up by bottlenecks within Adabas
- The efficiency of Adabas I/O buffering
- The level and efficiency of format processing, an important factor in Adabas CPU usage
- The identity of users who may be holding excessive numbers of records

Insight Help Main Menu

To get a detailed explanation about the SNAP panel, you can invoke HELP by pressing the PF1 key.

A comprehensive HELP facility provides online information about all Insight functions and error conditions. The major categories of information are indicated on the HELP Main Menu panel shown below.

HELP requests normally result in this HELP Main Menu prompt. However, if an error condition is indicated when HELP is invoked, menu selection is bypassed and information specific to the particular error is immediately displayed.

```
      C A - I N S I G H T   H E L P   M A I N   M E N U

      CODE  TOPIC OR ACTION
      ----  -
      O -   OVERVIEW
      B -   BASIC PROCEDURES
      M -   MANIPULATING REQUESTS
      C -   COMMAND SUMMARY

      D -   DISPLAYING REQUEST DATA
      S -   SCROLLING REQUEST OUTPUT

      X -   'SNAP' EXPLANATION
      Z -   'SHOW' EXPLANATION
      A -   ADABAS FUNCTIONS
      E -   ERROR RESOLUTION

      L -   LEAVE HELP
      . -   TERMINATE CA-INSIGHT
      ----  -
CHOICE:  _
```

Type the letter x (for SNAP Explanation) in the CHOICE field and press ENTER.

SNAP Display Help Panel Menu

The SNAP explanation panel is displayed.

S N A P D I S P L A Y E X P L A N A T I O N

CODE	INFORMATION CATEGORY
----	-----
G -	GENERAL INFORMATION
T -	TOP LINE OF DISPLAY
C -	COMMAND PROCESSING AREA
U -	USERS AREA
B -	I-O BUFFER POOL AREA
W -	NUCLEUS WORK POOLS AREA
Q -	HOLD QUEUE AREA
M -	HELP MAIN MENU
L -	LEAVE HELP
.	TERMINATE CA-INSIGHT

CHOICE: _

If you wanted help on the significance on I/O buffer pools, you would type the letter **b** in the CHOICE field.

SNAP Display Help Detail Panel

The first screen of I/O Buffer Pool Area help is displayed:

```
          I/O B U F F E R   A R E A   O F   ' S N A P '   D I S P L A Y
-----
BP SIZE:  1,156,352
LOG.IO:   636
PHY.IO:   141
EFF:      4.5
FLUSHES:  1
-----
BP SIZE:  ACTUAL NO. BYTES IN THE ADABAS I/O BUFFER POOL.

LOG.IO:  NO. LOGICAL ACCESSES TO ASSO AND DATA BLOCKS DURING THE 'SNAP'
          INTERVAL (OR DURING SESSION IN FIRST SNAP AFTER ADD OF 'SNAPSHOT').

PHY.IO:  NO. PHYSICAL I/OS TO ASSO AND DATA DURING THE 'SNAP' INTERVAL.

EFF:     BUFFER EFFICIENCY - NO. OF LOGICAL ACCESSES (ABOVE) DIVIDED BY NO.
          OF ASSO AND DATA PHYSICAL I/OS DURING THE 'SNAP' INTERVAL (ABOVE).

FLUSHES: BUFFER FLUSHES - NO. OF TIMES DURING THE 'SNAP' INTERVAL THAT THE
          NUCLEUS PAUSED TO REWRITE TO DISK ALL ASSO AND DATA BLOCKS THAT HAD
          BEEN MODIFIED WHILE IN THE BUFFER POOL.          (<ENTER> TO CONTINUE)
-----
OPTION:  N - NEXT SCREEN;   S - SNAP HELP MENU;   M - HELP MAIN MENU   ==> N
```

Press ENTER to walk through the sequence of related HELP panels, when such a sequence is present.

```

      I/O B U F F E R   A R E A   O F   ' S N A P '   D I S P L A Y
-----
                        %AVAIL:          98.0
                        %WRITE:           0.0
                        %ACTIVE:          0.0
-----

THESE 'INSTANTANEOUS' VALUES SHOW HOW THE I/O BUFFER POOL IS BEING USED AT THE
TIME THE 'SNAP' COMMAND IS EXECUTED.

%AVAIL:  % OF TOTAL BYTES USABLE TO HOLD ASSO AND DATA BLOCKS; ALWAYS LESS
          THAN 100% DUE TO LESS THAN PERFECT FIT OF BLOCKS INTO ELEMENTS.

%WRITE:  % OF TOTAL BYTES CONTAINING BLOCKS THAT HAVE BEEN MODIFIED AND
          MUST BE REWRITTEN TO DISK BEFORE THEY MAY BE OVERLAID IN THE POOL.
          (MAXIMUM FOR THIS CAN BE RAISED OR LOWERED BY Adabas ZAP OR PARM.)

%ACTIVE: % OF TOTAL BYTES CONTAINING BLOCKS THAT ARE CURRENTLY INVOLVED IN
          PROCESSING A COMMAND.  USUALLY VERY LOW.

                                           (<ENTER> TO CONTINUE)
-----
OPTION:  N - NEXT SCREEN;   S - SNAP HELP MENU;   M - HELP MAIN MENU   ==> N
-----
```

You can return at any time to the SNAP HELP menu by entering the letter **s** or the HELP Main Menu by entering the letter **m** in the OPTION field.

ADAMENU - The Adabas Functions Menu

The Adabas Functions Menu is a convenient point from which to invoke the functions available from Data Collectors that are executing with Adabas MPMs.

```

C A - I N S I G H T   A D A B A S   F U N C T I O N S   M E N U
SPECIFY NEXT FUNCTION WITH MENU CODE, PF-KEY, OR CA-INSIGHT COMMAND

CODE  FUNCTION                                           (COMMAND )
-----
 1  ADABAS SESSION HIGH WATER MARKS                     (HIWATER )
 2  ADABAS HOLD QUEUE SUMMARY BY FILE                   (HQFILSUM)
 3  LIST USERS OF A SPECIFIED FILE                       (UQEBYFIL)
 4  LIST ALL OR SUBSETS OF ADABAS USER QUEUE            (UQELIST )
 5  ADABAS USER AND COMMAND QUEUES COUNTS               (COUNTS )
 6  ADABAS I/O BUFFER POOL USAGE BY FILE               (IOBPSUM )
 7  ADABAS THREADS STATUS SUMMARY                      (THREADSM)
 8  ADABAS THREADS DETAILS                             (THREADET)
 9  LIST USERS WITH MOST RECORDS HELD                   (HQTOPUSR)
 A
 B  ADARUN PARAMETER VALUES                           (ADARUN  )
 C  SELECT USERS FROM A LIST FOR STOPU                  (STOPU   )
 D  ISSUE OPERATOR COMMANDS TO ADABAS MPM              (OPERCMD )
CODE: _

CMD: _____ REQ: _____ DBID: _____ SMPID: 0 DT: 02-11-21 TM: 23:03:06.7
PFK: 1=HELP 3=MENU 4=SELECT 5=SHOW 6=DISPLAY 9=SNAP 12=REQMGT INADAMNU

```

Each of these functions can also be invoked from any other Insight panel by using the Insight command name that is shown with its entry on this menu panel.

High Water Marks Panel

The Adabas Session High Water Marks (HIWATER) panel shows the allocated sizes and maximum use so far during a session of selected Adabas internal areas.

```

ADD      HIWATER WAS DONE ON DBID: 00003
          A D A B A S   S E S S I O N   H I G H - W A T E R   M A R K S

TOTAL USER SESSIONS:          15                                DBID: 00003

                                USERID          QUANTITY
                                -----
USER SESSION WITH MAX CMDS    ABEJU01          277
USER SESSION WITH MAX I/O     ABEJU01          234
USER SESSION WITH MAX CPU     ABEJU01           8 MINS   24 SECS

AREA                          ADARUN PARM      ALLOCATED      MAX USED      % USED
-----
USER QUEUE                    21 NU          9,380         2,144         22.86
COMMAND QUEUE                 20 NC          3,840          192           5.00
HOLD QUEUE                    10,000 NH       280,056         84           0.03
WORK POOL                     500,000 LWP     499,744       292,060        58.44
INTERNAL FORMAT POOL          200,000 LFP     200,000         1,344          0.67
TABLE OF SEQ. CMDS.           30,000 LQ        30,000          496           1.65
TABLE OF ISN LIST IDS         10,000 LI        10,000           0            0.00
SECURITY POOL                 15,000 LCP       15,000           0            0.00
ATTACHED BUFFERS              100 NAB        409,600       34,560          8.44
-----
CMD: _____ REQ: _____ DBID: 0 SMPID: 0 DT: 02-11-21 TM: 23:03:59.3
PFK: 1=HELP 3=ADAMENU 4=SELECT 5=SHOW 6=DISPLAY 9=SNAP INHIWATR

```

In addition, the number of user sessions so far and information about maximum activity by individual users are shown. This information can help a database administrator to see the need to increase the sizes of areas to avoid shortages or for performance improvements, or to decrease areas where current sizes appear excessive.

Press the PF3 key or type **ADAMENU** in the CMD field and press ENTER to return to the Adabas Functions Menu.

User Queue Panel

The User Queue (UQELIST) panel lists information about users that are currently reflected in the Adabas User Queue.

A D A B A S U S E R Q U E U E D I S P L A Y							
TERM-ID	TERM-ID IN HEX	INTRNLID	JOBNAME	HOLDS	THD-TIM	CMD5	I/O'S
TSU02807	E3E2E4F0F2F8F0F7	00000010	ABEJU01	0	.000	33	0
TSU02807	E3E2E4F0F2F8F0F7	0000000E	ABEJU01	0	.000	863	657
TSU02807	E3E2E4F0F2F8F0F7	0000000F	ABEJU01	0	.000	724	621

YOU MAY MARK -1- USER: D=DETAILS H=HOLDS S=STOPU *** END ***
 CMD: UQELIST REQ: _____ DBID: 0 SMPID: 0 DT: 02-11-21 TM: 23:06:15.7
 PFK: 1=HELP 3=ADAMENU 4=SELECT 5=SHOW 6=DISPLAY 9=SNAP INUQLIST

A user shown on this panel may be marked to select additional information or, under MVS only, to issue a STOPU operator command for the selected user.

With the REQ: field blank, all users in the User Queue are displayed. With the REQ: field non-blank, only those users whose user-id values begin with the value in the REQ: field is displayed. For instance, with REQ: equal to the value TSO, only the TSO users would be displayed.

Press the PF3 key or type **ADAMENU** in the CMD field and press ENTER to return to the Adabas Functions Menu.

Hold Queue Summary Screen

The Hold Queue Summary (HQQFILSUM) panel displays information only for those files that are currently reflected in the Adabas Hold Queue.

A D A B A S H O L D Q U E U E S U M M A R Y											
FNR	USERS	HOLDS	FNR	USERS	HOLDS	FNR	USERS	HOLDS	FNR	USERS	HOLDS
3	4	9	10	30	27	11	8	4	12	15	35
22	45	108	23	6	3	240	1	6	252	12	18

CMD:	REQ:	DBID:	0	SMPID:	0	DT:	02-11-21	TM:	23:07:05.7		
PFK:	1=HELP	3=ADAMENU	4=SELECT	5=SHOW	6=DISPLAY	9=SNAP	INHQSUM				

If more detailed information is desired for a given file, that file number can be specified in the REQ: field with the UQEBYFIL command. The UQEBYFIL display lists each user that is currently holding records in the specified file, with the number of records that user is holding in the specified file and the total number of records held by that user in any files.

Return to ADAMENU.

I/O Buffer Pool Panel

The I/O Buffer Pool (IOBPSUM) panel shows how the Adabas I/O buffer pool is currently being shared by Adabas files.

A D A B A S I / O B U F F E R P O O L - F I L E U S A G E												
BLOCK COUNTS:				AC	UI	MI	NI	TOTAL ASSO	TOTAL DATA	TOTAL ALL	% OF POOL	
FNR	FCB	FDT	DSST									
ALL	3	3	0	9	3	7	13	38	29	67	94.16	
10	1	1	0	6	1	4	4	17	1	18	18.31	
11	1	1	0	2	1	2	8	15	27	42	68.61	
19	1	1	0	1	1	1	1	6	1	7	7.24	

CMD: _____ REQ: _____ DBID: 0 SMPID: 0 DT: 02-11-21 TM: 23:08:14.6
 PFK: 1=HELP 3=ADAMENU 4=SELECT 5=SHOW 6=DISPLAY 9=SNAP INIOBPSM

This kind of information can give useful insights into the general kinds of activity of current users. For instance, very high numbers of index blocks for a given file could indicate expensive (and possibly inefficient) searches if only one user is active for the file, whereas if many users are active for the file, efficient but highly random search activity might be indicated.

Return to ADAMENU.

Stop Users Panel

The Stop Users (STOPU) panel lists currently active Adabas users that can be stopped, although only under MVS.

M A R K A D A B A S U S E R S F O R S T O P U							
TERM-ID	TERM ID IN HEX	INTRNLID	JOBNAME	HOLDS	UQEUSER	LAST ACTIVITY	
-----	-----	-----	-----	-----	-----	-----	
TSU02807	E3E2E4F0F2F8F0F7	00000010	ABEJU01	0		20030121	230859
TSU02807	E3E2E4F0F2F8F0F7	0000000E	ABEJU01	0	ABEJU01	20030121	230359
TSU02807	E3E2E4F0F2F8F0F7	0000000F	ABEJU01	0	NùÇITnz?	20030121	230859

*** END ***

CMD: STOPU REQ: DBID: 0 SMPID: 0 DT: 02-11-21 TM: 23:08:59.1
 PFK: 1=HELP 3=ADAMENU 4=SELECT 5=SHOW 6=DISPLAY 9=SNAP INSTOPUP

Optionally, one or multiple users shown on a given panel may be marked for issuing STOPU operator commands. After the ENTER key is pressed following marking, the marked users are displayed one at a time for confirming or abandoning the STOPU for the indicated user. Unicenter CA-APAS security allows the issuing of STOPU operator commands to be restricted to specified individual users or groups of users.

Return to ADAMENU.

Operator Command Panel

This Operator Command (OPERCMD) panel shows the format of Adabas operator commands that Unicenter CA-APAS can issue, although only under MVS.

I S S U E	O P E R A T O R	C O M M A N D S	T O	A D A B A S
DCQ	TNAA=SSSSSS	READONLY=YES/NO		DLOCKF
DDIB	TNAE=SSSSSS	UTIONLY=YES/NO		DFILES=FFFFF
DHQ	TNAX=SSSSSS			DFILUSE=FFFFF
DHQ	TT=SSSSSS	DDSF		(UN)LOCKF=FFFFF
DHQA	CT=SSSSSS	DAUQ		(UN)LOCKU=FFFFF
DNC		DUUQE		(UN)LOCKX=FFFFF
DNH	RDUMPST	DELUF=FFFFF		
DNU	(NO)LOGUX	DELUI=SSSSSS		FE0FCL
DTH	(NO)LOGFB	DUQE=X'USERID'		FE0FPL
DUQ	(NO)LOGIB			SYNCC
DUQA	(NO)LOGIO	STOPF=FFFFF		
DPARM	(NO)LOGRB	STOPI=SSSSSS		ADAEND
DRES	(NO)LOGSB	STOPU=X'USERID'		CANCEL
DSTAT	(NO)LOGVB	STOPU=JOBNAME		HALT

OPERATOR CMD: _____

TYPE ANY VALID, COMPLETE ADABAS OPERATOR COMMAND ABOVE

CMD: _____ REQ: _____ DBID: 0 SMPID: 0 DT: 02-11-21 TM: 23:09:43.0
 PFK: 1=HELP 3=ADAMENU 4=SELECT 5=SHOW 6=DISPLAY 9=SNAP INOPRCMD

This facility eliminates the need for a database administrator to contact a computer operator or leave the Insight session in order to send such commands to an active Adabas MPM. The Unicenter CA-APAS optional security feature makes it possible to restrict use of this operator command facility to specified individual users and groups of users.

Exit Insight by typing a period in the CMD field and pressing ENTER.

Glossary

ACBX

Adabas Control Block Extension (ACBX) facility. This Unicenter CA-APAS facility uses exits from Adabas link routines to develop accounting and performance data not otherwise available for individual Adabas commands.

Batch Mode

An optional means to execute the Data Collector. The Data Collector reads Adabas Command Log records from a standard Adabas Command Log file or from an Unicenter CA-APAS COPY file, producing requested print-formatted reports or machine-readable files of specified data fields or both.

Command Data Requests

Statements that specify what information is desired about the processing of Adabas commands. Unicenter CA-APAS includes numerous default requests; users may modify these or develop entirely new ones using a powerful and flexible statement syntax. The statements allow the identification of the data fields desired, overriding formats or headings, etc., formatting output, and conditional command-selection logic.

COPY Files

An optional type of output from the Data Collector. The Data Collector writes all or selected Adabas Command Log records, with additional fields derived by Unicenter CA-APAS, to COPY files. COPY files offer a more flexible and efficient alternative to Command Log files written by the Adabas nucleus.

Data Collector

The primary component used with Unicenter CA-APAS and/or Unicenter CA-SpaceMan. Its basic purpose is to do selective processing and reporting of data obtained from the Adabas Command Logging process or data obtained from the CA-SpaceMan IOLOG facility. It consists of a variable number of individual program modules and may be executed in batch mode or MPM mode.

Dual Output Files

An optional means of handling the Unicenter CA-APAS output to any of the files it writes. Dual file processing enables output to be continued throughout long-running Adabas nucleus sessions without dedication of tape drives or allocation of excessively large direct-access files. This method involves the use of two physical direct-access files for a given logical file, allowing output to be directed to one file of the pair while data from the other is being copied. Unicenter CA-APAS fully supports use of dual files with provisions to submit the required copy jobs and the program to do the copy processing.

ENQ-TIME

A special command data field derived using the Unicenter CA-APAS ACBX facility. ENQ-TIME for a command is the period begun when the command is issued by the application program and ending when the Adabas nucleus begins processing the command in an Adabas thread. ENQ-TIME provides the only way to precisely measure the extent to which update, complex search, or simple commands are being delayed due to shortages of related Adabas resources. These types of delays sometimes can cause additional delays in use of COM-LETE processing threads, compounding the impact on overall system throughput.

ENSU-BUFFER

A main-memory buffer used to temporarily hold Adabas Command Log record images until the Data Collector sub-task modules are ready to process them.

MPM Mode

An optional way to execute the Data Collector. The Data Collector processes Adabas Command Log records as they are passed via the Adabas user exit 4. In this mode, the Data Collector can support real-time Adabas monitoring from Unicenter CA-APAS and can produce requested print-formatted reports or machine-readable files of specified data fields or both. This mode optionally allows all Unicenter CA-APAS reporting capabilities to be used without having physical Command Logs written by Adabas.

RABN-RANGES

An optional reporting facility of Unicenter CA-APAS whereby users may define logical groupings of Relative Adabas Block Numbers (RABNs) and then have Unicenter CA-APAS report Adabas physical I/O counts to each defined range.

SNAPSHOT

A special Unicenter CA-APAS command data request that can be used by Unicenter CA-APAS to report about selected aspects of Adabas processing on a real-time basis.

Unicenter CA-SpaceMan

Unicenter CA-SpaceMan provides capabilities for reporting and controlling DASD space allocated to Adabas databases. CA-SpaceMan is integrated with Unicenter CA-APAS.

Index

A

Abend, 4-18
Abnormal termination, 2-9, 4-18
ACBX, 2-14
Adabas Command Log, 1-3
Adabas Control Block Extension, 2-14
Adabas Functions Menu, A-25
Adabas High Water Marks, A-26
Adabas Hold Queue, 3-8, 4-21
Adabas User Queue, A-27
ADAM, 3-15
ADD, 4-9, 4-13, 4-14
ALLTRACE request, 3-5

B

Batch jobs, 3-13
Batch mode, 2-2
Bottlenecks, 3-12, 4-21
Browsing saved requests, 4-8
BY clause, 2-5, 2-9
 omitting, 2-4
BY field, 4-20

C

CHECK, 4-9, 4-13, 4-14
CLXTRACE, 4-22
CMD field, 4-4
Column headings, 2-12, 2-13
Command data requests, 4-4, 4-19
Command Log records, 2-14
Command subsets, 2-4, 2-7
Commands
 entering, 4-4
Complex searches, 3-12, 4-21
COPY
 files, 1-3, 4-15
 requests, 1-3
 statement, 2-14
CPU bound, 3-14
CPU time, 4-10
CPU utilization, 4-6
Creating a new request, A-15
CTYPLOAD, 4-21

D

Data fields, 2-3, 2-4
DBGIOR5
 I/O logging, 4-15
 module, 4-15
DBID field, 4-5, 4-7

DDNAME, 2-11, 2-12

DECLARE statement, 2-6

Defining

data fields, 2-6

user-defined fields, 2-6

Defining output, 2-11

DELBYNAM command, 4-10

DELBOWN command, 4-10

DELETE, 4-10

Deleting requests, A-13

DISPLAY, 4-12, 4-13, 4-14

Displaying a new request, A-19

Displaying data, 4-12

DSTAT, 3-12

DUAL files, 2-14

Dual output files, 4-15

E

Editor panel, A-18

ENQUEUE, 3-12, 4-21

Entering commands, 4-4

Entering operands, 4-4

Errors Reported by Natural, 4-16

Exception, 4-7

Exception reporting, 3-2, 4-22

EXTRACT statement, 2-3, 2-11

F

Field length and format, 2-12

FILLOAD panel, A-7

FILTRACE, 4-22

FLIPXY, 4-15

G

Generate Trace Request panel, A-8

Generating requests, 4-7

Global variables, 4-6

Grand Totals, 2-4

H

HD parameter, 2-12

HELP facility, 4-5

HIWATER, A-26

Hold logic, 3-9

Hold Queue Summary, A-28

HOLD-COUNT field, 3-9

HOLDQ request, 3-8

HQFILSUM panel, A-28

I

I/O bound, 3-14

I/O Buffer Pool, A-29

INQUIT, 4-6

Insight Help main menu, A-21

Insight panel, A-2

INSIGHT-LINES parameter, 2-11

INTERVAL parameter, 2-9

Invalid commands or operands, 4-17

IOBPSUM panel, A-29

IOLOGSW command, 4-15

ISN-QUANTITY, 3-7

J

JCL, 2-11, 2-12

L

L3DEUSE, 3-15
LOG, 1-3
Logical sequential reading, 3-15
Looping, 3-9
LS, 4-18

M

Machine-readable data files, 2-11
Main Menu panel, A-3
Manipulating Unicenter CA-APAS requests, 4-7
Modes of Execution, 2-2
MPM mode, 2-2, 2-14
Multifetch, 3-14

N

NATLOAD panel, A-12
Natural commands, 4-5, 4-9
Natural editor screen, A-10
Natural operands, 4-5

O

Online display, 2-11
Operands
 entering, 4-4
Operator Command, A-30, A-31
OPERCMD panel, A-31
OUTPUT-FILE parameter, 2-11
Override column headers, 2-13, 4-20
Override lengths or formats, 2-12
OWNER keyword, 2-15

P

PAUSE, 4-10
Peak processing times, 3-9
Performance History System, 6-1
Performance limits, 3-2
Predefined requests, 4-11
Prefetch, 3-14
PREFLSB, 3-14
Print record headers, 2-11
PRINT TO parameter, 2-11, 2-12
Print-formatted reports, 2-12
Program profile, 3-6
Program trace, 4-22
PS, 4-18

Q

Quantity of report lines, 4-20

R

Rank, 3-11
REDEFINES, 2-6
Report Dimensions, 4-19
Report line
 width, 4-19
 width exceeded, 4-17
REQ field, 4-5
Request
 ownership, 2-15
 security, 2-15
 status, 4-11
Request Management panel, A-12
RESET, 4-9
Resetting request data, 4-9
Resolving errors, 4-16

Response Codes, 3-3
Response times, 3-13
RESPONSE-CLASS, 3-3
Restarting requests, 4-10
Restricted commands, 3-5
RESUME, 4-10
RSPCODES, 3-3, 4-22

S

Scrolling data, 4-13
Scrolling errors, 4-17
Search logic, 3-7
SECURITY keyword, 2-15
SELECT, 4-8, A-4, A-5
Selection of saved requests, 4-8
Sensitive files, 3-4
Shortage of threads, 3-12
SHOW, 2-9, 4-11, A-6
SIGRESP, 3-3
SNAP command, 3-8, 4-14, 4-21
SNAP Display Help detail panel, A-23
SNAP Display Help panel menu, A-22
SNAPSHOT panel, A-20
SNAPSHOT request, 3-8, 4-11, 4-14, 4-21
SORT DESC, 2-9, 3-11, 4-20, 4-21
Source code work area, 4-9
Sources of workloads, 3-11
Stacking fields in a column, 2-13
START, 4-8
START request VERZAPS, 4-16
START request ZAPDECL, 4-16
Starting saved requests, 4-8
Starting value, 4-8
Stop Users, A-30

STOPU panel, A-30
Submitting request statements, 2-2
SUBTOT=OFF, 2-5, 4-20
Subtotals, 2-4
SUMMARIZE statement, 2-4, 2-11
Summary, 4-7
Summary functions, 2-3, 2-4
Suspending requests, 4-10
Syntax checking, 4-9, A-17
Syntax errors, 4-17

T

Terminal screen size errors, 4-18
Terminating a Unicenter CA-APAS session, 4-6
THRDCMDS, 3-14
THRDUSRS, 3-13
Thread 1 saturated, 4-21
THREADS, 3-12
Time intervals, 2-9
TITLE clause, 4-20
Trace, 4-7
 program, 2-3
 program trace, 3-5
TRACE requests, 4-22
Transaction Time Limit, 3-8
TT, 3-8

U

Unicenter CA-APAS
 capabilities, 4-2
 conducting a session, 4-3
 HELP facility, 4-5
 interrupted session, 4-5
 Overview, 4-1
 Performance History System, 6-1
 Requests, 4-19

terminating a session, 4-6
UNLOG, 1-3
Update commands, 3-12
UPDSAT, 3-9, 3-12
UQELIST panel, A-27
USER1, 2-6
USER2, 2-6
USER3, 2-6
USERA, 2-6
USERB, 2-6
USERC, 2-6
User-exit routine, 2-6

V

Virtual storage, 2-5, 4-10

W

WHERE clause, 2-7, 3-2, 4-20

Workloads, 3-11

Z

ZAPs, 4-16

